

日本国特許庁
PATENT OFFICE
JAPANESE GOVERNMENT

JC931 U.S. PRO
09/702755
11/01/00


別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日

Date of Application:

1999年11月17日

出願番号

Application Number:

平成11年特許願第327178号

出願人

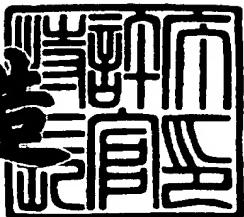
Applicant (s):

株式会社東芝

2000年 9月18日

特許庁長官
Commissioner.
Patent Office

及川耕造



出証番号 出証特2000-3075654

【書類名】 特許願
【整理番号】 A009905638
【提出日】 平成11年11月17日
【あて先】 特許庁長官 殿
【国際特許分類】 G06F 15/00
【発明の名称】 コンピュータシステムおよびデータ転送制御方法
【請求項の数】 14

【発明者】

【住所又は居所】 東京都青梅市末広町2丁目9番地 株式会社東芝青梅工
場内

【氏名】 中村 伸隆

【特許出願人】

【識別番号】 000003078

【氏名又は名称】 株式会社 東芝

【代理人】

【識別番号】 100058479

【弁理士】

【氏名又は名称】 鈴江 武彦

【電話番号】 03-3502-3181

【選任した代理人】

【識別番号】 100084618

【弁理士】

【氏名又は名称】 村松 貞男

【選任した代理人】

【識別番号】 100068814

【弁理士】

【氏名又は名称】 坪井 淳

【選任した代理人】

【識別番号】 100092196

【弁理士】

【氏名又は名称】 橋本 良郎

【選任した代理人】

【識別番号】 100091351

【弁理士】

【氏名又は名称】 河野 哲

【選任した代理人】

【識別番号】 100088683

【弁理士】

【氏名又は名称】 中村 誠

【選任した代理人】

【識別番号】 100070437

【弁理士】

【氏名又は名称】 河井 将次

【手数料の表示】

【予納台帳番号】 011567

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 コンピュータシステムおよびデータ転送制御方法

【特許請求の範囲】

【請求項1】 バストランザクションの伝達に必要なデータを第1および第2のコントローラ間で授受するコンピュータシステムにおいて、

送信対象の複数のデータそれぞれに連続するIDを付加し、それら複数のデータを、受信側からの肯定応答の返信を待たずに、送信側のコントローラから受信側のコントローラに順次送信するデータ送信手段と、

データを受信する度に、その受信したデータに対応するIDが付加された肯定応答を、受信側のコントローラから送信側のコントローラに送信する肯定応答送信手段と、

送信済みのデータ毎に受信側のコントローラからの肯定応答の返信の有無を管理し、前記受信側のコントローラから再送要求を受けた場合、前記肯定応答の返信待ちのデータを、ID順に再送する手段とを具備することを特徴とするコンピュータシステム。

【請求項2】 前記受信側のコントローラは、前記送信側コントローラから送信されるデータを受信することができないとき、あるいは受信したデータのIDが不連続であるときに、前記再送要求を前記送信側のコントローラに送信することを特徴とする請求項1記載のコンピュータシステム。

【請求項3】 前記第1および第2のコントローラ間のデータ転送にエラーが生じたとき、予め決められたビットパターンの送受信を前記第1および第2のコントローラ間で実行することによって前記第1および第2のコントローラ間の通信路の同期を再確立し、受信側のコントローラから送信側のコントローラに受信済みのデータのIDを通知した後に、前記第1および第2のコントローラを通常動作状態に復帰させるエラー回復手段をさらに具備し、

通常動作状態に復帰した場合、前記データ送信手段は、受信側のコントローラから通知された受信済みのデータの次のデータから、送信処理を再開することを特徴とする請求項1記載のコンピュータシステム。

【請求項4】 前記送信側のコントローラは、前記受信側からの肯定応答の

IDが不連続であるときに、前記第1および第2のコントローラ間のデータ転送にエラーが生じたことを検出することを特徴とする請求項3記載のコンピュータシステム。

【請求項5】 バストランザクションの伝達に必要なデータを第1および第2のコントローラ間で授受するコンピュータシステムにおいて、

送信待ちのデータそれぞれに連続するIDを付加し、予め決められた複数個のデータを、受信側からの肯定応答の返信を待たずに、送信側のコントローラから受信側のコントローラに連続的に送信するデータ送信手段と、

データを受信する度に、その受信したデータに対応するIDが付加された肯定応答を、受信側のコントローラから送信側のコントローラに送信する肯定応答送信手段と、

前記受信側のコントローラからの肯定応答の返信の有無を管理し、前記肯定応答の返信待ちのデータについては繰り返し送信されるように、前記肯定応答の返信待ちのデータを含めて、前記複数個のデータ単位での連続送信処理を、前記データ送信手段に繰り返し実行させる手段とを具備することを特徴とするコンピュータシステム。

【請求項6】 前記第1および第2のコントローラ間のデータ転送にエラーが生じたとき、予め決められたビットパターンの送受信を前記第1および第2のコントローラ間で実行することによって前記第1および第2のコントローラ間の通信路の同期を再確立し、受信側のコントローラから送信側のコントローラに受信済みのデータのIDを通知した後に、前記第1および第2のコントローラを通常動作状態に復帰させるエラー回復手段をさらに具備し、

通常動作状態に復帰した場合、前記データ送信手段は、受信側のコントローラから通知された受信済みのデータの次のデータから、送信処理を再開することを特徴とする請求項5記載のコンピュータシステム。

【請求項7】 前記送信側のコントローラは、前記受信側からの肯定応答のIDが不連続であるときに、前記第1および第2のコントローラ間のデータ転送にエラーが生じたことを検出することを特徴とする請求項6記載のコンピュータシステム。

【請求項8】 前記第1および第2のコントローラは、ホスト装置とその機能拡張のための拡張ユニットとにそれぞれ分散して設けられており、前記ホスト装置側のバスと前記拡張ユニット側のバスとの間を、前記コントローラ間に配設されるシリアル伝送路を介して接続するための単一のブリッジ装置として機能するものであることを特徴とする請求項1乃至7のいずれか1項記載のコンピュータシステム。

【請求項9】 ホスト装置とその機能拡張のための拡張ユニットとにそれぞれ分散して設けられた物理的に異なる第1および第2のコントローラから構成され、前記ホスト装置側のバスと前記拡張ユニット側のバスとの間を、前記コントローラ間に配設されるシリアル伝送路を介して接続するためのブリッジ装置を有するコンピュータシステムであって、

前記第1および第2のコントローラの各々は、

送信対象の複数のデータそれぞれに連続するIDを付加し、それら複数のデータを、受信側からの肯定応答の返信を待たずに、送信側のコントローラから受信側のコントローラに順次送信するデータ送信手段と、

データを受信する度に、その受信したデータに対応するIDが付加された肯定応答を、受信側のコントローラから送信側のコントローラに送信する肯定応答送信手段と、

送信済みのデータ毎に受信側のコントローラからの肯定応答の返信の有無を管理し、前記受信側のコントローラから再送要求を受けた場合、前記肯定応答の返信待ちのデータを、ID順に再送する手段とを具備することを特徴とするコンピュータシステム。

【請求項10】 ホスト装置とその機能拡張のための拡張ユニットとにそれぞれ分散して設けられた物理的に異なる第1および第2のコントローラから構成され、前記ホスト装置側のバスと前記拡張ユニット側のバスとの間を、前記コントローラ間に配設されるシリアル伝送路を介して接続するためのブリッジ装置を有するコンピュータシステムであって、

前記第1および第2のコントローラの各々は、

送信待ちのデータそれぞれに連続するIDを付加し、予め決められた複数個の

データを、受信側からの肯定応答の返信を待たずに、送信側のコントローラから受信側のコントローラに連続的に送信するデータ送信手段と、

データを受信する度に、その受信したデータに対応するIDが付加された肯定応答を、受信側のコントローラから送信側のコントローラに送信する肯定応答送信手段と、

前記受信側のコントローラからの肯定応答の返信の有無を管理し、前記肯定応答の返信待ちのデータについては繰り返し送信されるように前記肯定応答の返信待ちのデータを含めて、前記複数個のデータ単位での連続送信処理を、前記データ送信手段に繰り返し実行させる手段とを具備することを特徴とするコンピュータシステム。

【請求項11】 送信対象の複数のデータそれぞれに連続するIDを付加し、それら複数のデータを、受信側からの肯定応答の返信を待たずに、送信側から受信側に順次送信するデータ送信手段と、

データを受信する度に、その受信したデータに対応するIDが付加された肯定応答を、受信側から送信側に送信する肯定応答送信手段とを具備し、

送信側から受信側へのデータの送信と、受信側から送信側への肯定応答の返信とを非同期で行うように構成されていることを特徴とするコンピュータシステム。

【請求項12】 コンピュータシステムに設けられた第1および第2のコントローラ間で、バストランザクションの伝達に必要なデータを授受するためのデータ転送制御方法であって、

送信対象の複数のデータそれぞれに連続するIDを付加し、それら複数のデータを、受信側からの肯定応答の返信を待たずに、送信側のコントローラから受信側のコントローラに順次送信し、

データを受信する度に、その受信したデータに対応するIDが付加された肯定応答を、受信側のコントローラから送信側のコントローラに送信し、

送信済みのデータ毎に受信側のコントローラからの肯定応答の返信の有無を管理し、前記受信側のコントローラから再送要求を受けた場合、前記肯定応答の返信待ちのデータを、ID順に前記送信側のコントローラから前記受信側のコント

ローラに再送することを特徴とするデータ転送制御方法。

【請求項13】 コンピュータシステムに設けられた第1および第2のコントローラ間で、バストランザクションの伝達に必要なデータを授受するためのデータ転送制御方法であって、

送信待ちのデータそれぞれに連続するIDを付加し、予め決められた複数個のデータを、受信側からの肯定応答の返信を待たずに、送信側のコントローラから受信側のコントローラに連続的に送信し、

データを受信する度に、その受信したデータに対応するIDが付加された肯定応答を、受信側のコントローラから送信側のコントローラに送信し、

前記受信側のコントローラからの肯定応答の返信の有無を管理し、前記肯定応答の返信待ちのデータについては繰り返し送信されるように前記肯定応答の返信待ちのデータを含めて、前記複数個のデータ単位での連続送信処理を、繰り返し実行することを特徴とするデータ転送制御方法。

【請求項14】 送信対象の複数のデータそれぞれに連続するIDを付加し、それら複数のデータを、受信側からの肯定応答の返信を待たずに、送信側から受信側に順次送信するデータ送信ステップと、

データを受信する度に、その受信したデータに対応するIDが付加された肯定応答を、受信側から送信側に送信する肯定応答送信ステップとを具備し、

送信側から受信側へのデータの送信と、受信側から送信側への肯定応答の返信とを非同期で行うことを特徴とするデータ転送制御方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明はコンピュータシステムおよびデータ転送制御方法に関し、特にバストランザクションを伝達するために必要なデータを正確且つ速やかに転送することができるよう改良されたコンピュータシステムおよびデータ転送制御方法に関する。

【0002】

【従来の技術】

近年、携行が容易でバッテリにより動作可能なノートブックタイプのパーソナルコンピュータ（以下、ノートPCと称する）が種々開発されている。ノートPCの中には、その機能拡張のために、拡張ユニットに必要に応じて装着できるよう構成されているものがある。ノートPC本体から拡張ユニットのリソースを有効利用できるようにするためには、ノートPC本体のバスと拡張ユニット内のバスとを接続することが重要である。このバス接続により、拡張ユニット内のバス上のデバイスをノートPC本体内のデバイスと同様に扱うことが可能になる。

【0003】

多くのパーソナルコンピュータでは、PCIバス（Peripheral Component Interconnect Bus）が使用されている。したがって、ノートPC本体と拡張ユニットとの間のバス接続は、PCIバスの信号線群の数に相当する多数のピンを有するドッキング用コネクタをノートPC本体側と拡張ユニット側にそれぞれ設け、そのドッキング用コネクタを介して両者のPCIバスを物理的に接続することによって行うのが通常である。

【0004】

しかし、この構成では、ドッキング用コネクタの実装に多くの面積が必要とされるため、ノートPC本体の小型化・薄型化を図る上では不利である。さらに、ノートPC本体側と拡張ユニット側それぞれのコネクタ実装位置を合わせなければならぬため、新たな製品開発を行う上では、物理的な筐体構造に制約が加わることになる。

【0005】

【発明が解決しようとする課題】

そこで、最近では、ノートPC本体側のPCIバスと拡張ユニット側のPCIバスとの間をつなぐPCI-PCIブリッジを物理的に異なる第1および第2の2つのコントローラで構成し、それらコントローラ間の情報の授受をシリアル転送によって行うという技術が本出願人によって提案されている（特願平11-183919号明細書）。この場合、第1および第2の2つのコントローラはノートPC本体側と拡張ユニット側とに分かれて配置される。ノートPC本体側のPCIバスから拡張ユニット側のPCIバスへのトランザクションの伝達に必要なデ

ータは、第1のコントローラにてパラレルデータからシリアルデータに変換された後に、ケーブルなどのシリアル伝送線路を介して第2のコントローラに伝達される。そして、第2のコントローラではシリアルデータからパラレルデータへの変換が行われ、そして拡張ユニット側のPCIバス上にトランザクションが展開される。これにより、ノートPC本体と拡張ユニットとを信号線数の少ない細いケーブルで接続することが可能となる。

【0006】

しかし、このように単一のPCI-PCIブリッジを物理的に異なる第1および第2の2つのコントローラによって構成した場合には、システムの動作の信頼性を維持するため、これら2つのコントローラ間でデータ転送を確実に行うことが必要となる。さらに、2つのコントローラ間のデータ転送速度は、システム全体のスループットに大きく影響を与えることになる。よって、バストランザクションの伝達に必要なデータを2つのコントローラ間で正確、且つ迅速に授受するための新たなフロー制御の実現が要求されている。

【0007】

コンピュータや通信の分野におけるデータ転送の多くは、ハンドシェークを用いて行われている。受信側はエラーを検出したらNACK (Negative ACK) を送信側へ送る。送信側はNACKを受け取ったらそのメッセージを再送する。受信側はデータを正しく受け取ったならば、ACKを送信側へ送る。送信側は、ACKを受け取ると、次のデータの送信処理に進む。しかし、このような方式では、その実現は簡単であるものの、十分な通信速度を得ることは困難である。

【0008】

本発明は上述の事情に鑑みてなされたものであり、バス間でトランザクションを伝達するために必要なデータを2つのコントローラ間で正確、且つ迅速に転送することが可能な新たなフロー制御を実現し、十分なスループットと動作の信頼性の向上を図ることが可能なコンピュータシステムおよびデータ転送制御方法を提供することを目的とする。

【0009】

【課題を解決するための手段】

上述の課題を解決するため、本発明は、バストランザクションの伝達に必要なデータを第1および第2のコントローラ間で授受するコンピュータシステムにおいて、送信対象の複数のデータそれぞれに連続するIDを付加し、それら複数のデータを、受信側からの肯定応答の返信を待たずに、送信側のコントローラから受信側のコントローラに順次送信するデータ送信手段と、データを受信する度に、その受信したデータのIDと同一のIDが付加された肯定応答を、受信側のコントローラから送信側のコントローラに送信する肯定応答送信手段と、送信済みのデータ毎に受信側のコントローラからの肯定応答の返信の有無を管理し、前記受信側のコントローラから再送要求を受けた場合、前記肯定応答の返信待ちのデータを、ID順に再送する手段とを具備することを特徴とする。

【0010】

このコンピュータシステムにおいては、送信側のコントローラではデータに連続するIDが付加されて、受信側のコントローラに転送される。この場合、データの送信の度に受信側のコントローラからの肯定応答を待つ必要はなく、複数のデータそれぞれが、肯定応答を待たずに順次受信側のコントローラに転送されていく。受信側のコントローラは、基本的には、データを受信する度に、肯定応答を送信側に返す。ただし、この場合、肯定応答には受信したデータに対応するIDが付加される。これは、肯定応答の返信を、送信側からのデータの送信と非同期に実行できるようにするためである。よって、送信側は、受信側からの肯定応答を待たずに次のデータの送信を行うことが可能となる。

【0011】

さらに、送信側では、肯定応答に付加されたIDを用いることにより、送信済みのデータ毎に受信側のコントローラからの肯定応答の返信の有無を知ることができるので、通常のハンドシェークと同様に、個々のデータ毎にその受信の有無を正しく知ることができる。また、受信側がどのデータの受信まで完了しているかを知ることができるので、例えば受信側のバッファのオーバーフローなどによって受信側がデータを受信できなかった場合には、未受信のデータを再送することができる。よって、通信速度の向上と信頼性の向上との両立を図ることができ

る。

【0012】

したがって、以上のフロー制御の仕組みを、バスブリッジを構成する物理的に異なる2つのコントローラに適用することにより、バストランザクションの伝達に必要なデータを、迅速、且つ正確に転送することが可能となり、バスブリッジとしての動作の信頼性および性能を維持した状態で、ホスト装置と拡張ユニット間のシリアル接続を実現することができる。

【0013】

送信側では、送信済みのデータであっても、該当する肯定応答を受けるまでは肯定応答待ちのデータとして管理しておけばよい。

【0014】

また、本発明のコンピュータシステムは、前記第1および第2のコントローラ間のデータ転送にエラーが生じたとき、予め決められたビットパターンの送受信を前記第1および第2のコントローラ間で実行することによって前記第1および第2のコントローラ間の通信路の同期を再確立し、受信側のコントローラから送信側のコントローラに受信済みのデータのIDを通知した後に、前記第1および第2のコントローラを通常動作状態に復帰させるエラー回復手段をさらに具備し、通常動作状態に復帰した場合、前記データ送信手段は、受信側のコントローラから通知された受信済みのデータの次のデータから、送信処理を再開することを特徴とする。

【0015】

このように、エラー検出時には、通信路の同期の再確立から始めることにより、初期化時と同様、様々なエラーからの回復を確実に行うことが可能となる。さらに、エラー回復処理では、コントローラ間の論理的な不一致の解消を行うため、受信側から送信側に、受信済みのデータのIDが通知される。これにより、エラー回復後は、送信側のデータ送信手段は、受信側のコントローラから通知された受信済みのデータの次のデータから送信処理を正しく再開することができる。

【0016】

前述したように肯定応答にはIDが付加されているので、エラー検出は、受信

側からの肯定応答のIDの連続性をチェックすることによって容易に行うことができる。

【0017】

また、本発明のコンピュータシステムは、バストランザクションの伝達に必要なデータを第1および第2のコントローラ間で授受するコンピュータシステムにおいて、送信待ちのデータそれぞれに連続するIDを付加し、予め決められた複数個のデータを、受信側からの肯定応答の返信を待たずに、送信側のコントローラから受信側のコントローラに連続的に送信するデータ送信手段と、データを受信する度に、その受信したデータのIDと同一のIDが付加された肯定応答を、受信側のコントローラから送信側のコントローラに送信する肯定応答送信手段と、前記受信側のコントローラからの肯定応答の返信の有無を管理し、前記肯定応答の返信待ちのデータについては繰り返し送信されるように前記肯定応答の返信待ちのデータを含めて、前記複数個のデータ単位での連続送信処理を、前記データ送信手段に繰り返し実行させる手段とを具備することを特徴とする。

【0018】

このコンピュータシステムにおいては、送信待ちのデータのみならず、肯定応答の返信待ちのデータをも含めて、複数個のデータ単位での連続送信処理が繰り返し実行される。このため、受信側は自身のタイミングでデータを受信すれば良く、再送要求などを行わずとも、任意のタイミングでデータを受信することができる。よって、ややスループットは落ちるもの、再送処理のための制御が不要となり、制御の簡単化を実現できる。もちろん肯定応答を受けたデータについては送信対象から除外し、その代わりに、送信待ちのデータがあればそれを含めて送信すればよい。

【0019】

【発明の実施の形態】

以下、図面を参照して、本発明の実施形態を説明する。

図1には、本発明の一実施形態に係るコンピュータシステムの構成が示されている。このコンピュータシステムはノートブックタイプのパーソナルコンピュータ(PC)であり、そのPC本体100と、このPC本体100にケーブル接続し

て使用可能な機能拡張用のドッキングステーション200とから構成されている。ケーブル内の信号線は、図示のようにシリアル転送路300から構成されている。シリアル転送路300は、LVDS (Low Voltage Differential Signal) 線路、I²Cバス、およびその他のシリアル制御信号線から構成されている。

【0020】

LVDS線路は、PC本体100内のPCIバス2とドッキングステーション200内のPCIバス4との間をシリアル接続するために使用されるシリアル伝送路である。このシリアル伝送路を介した高速ピットシリアル信号転送によって、PC本体100のPCIバス2とドッキングステーション200内のPCIバス4との間のバストランザクションの受け渡しが行われる。

【0021】

PC本体100には、図示のように、プロセッサバス1、PCIバス2、ISAバス3、CPU11、ホスト-PCIブリッジ12、主メモリ13、表示コントローラ14、プライマリPCIシリアル転送コントローラ15、PCI-ISAブリッジ16、I/Oコントローラ17、各種ISAデバイス18、埋め込みコントローラ(EC)19、およびその他の各種コントローラ20等から構成されている。

【0022】

CPU11は本PCシステム全体の動作を制御するためのものであり、主メモリ13にロードされるオペレーティングシステム、システムBIOS、および他の各種プログラムを実行する。ホスト-PCIブリッジ12は、CPUバス1とプライマリPCIバス2を双方向で接続するブリッジ装置であり、ここには主メモリ13をアクセス制御するためのメモリコントロールロジック、および表示コントローラ14との接続に使用されるAGPの制御ロジックも内蔵されている。ホスト-PCIブリッジ12はプライマリPCIバス2上のバスマスターとして機能することが出来る。主メモリ13は、オペレーティングシステム、処理対象のアプリケーションプログラム/ユーティリティ、およびアプリケーションプログラム等によって作成されたユーザデータ等を格納する。

【0023】

プライマリPCIシリアル転送コントローラ15は、ドッキングステーション200内に設けられたセカンダリPCIシリアル転送コントローラ35と共同して論理的に1個のPCI-PCIブリッジ装置を構成するものである。このPCI-PCIブリッジ装置は、PC本体100内のPCIバス2とドッキングステーション200内のPCIバス4との間を双方向で接続するためのものであり、PCIバス2上のデバイスがPCIバス4上のデバイスをアクセスすること、およびその逆を可能にする。PCI-PCIブリッジ装置から見てホスト側に近い方のPCIバス2はPCI-PCIブリッジ装置のプライマリPCIバスとなり、遠い方のPCIバス4はPCI-PCIブリッジ装置のセカンダリPCIバスとなる。すなわち、本実施形態においては、プライマリPCIバス2とセカンダリPCIバス4をつなぐPCI-PCIブリッジ装置を、物理的に異なる2個のコントローラ（プライマリPCIシリアル転送コントローラ15、セカンダリPCIシリアル転送コントローラ35）に分割し、その間をLVDS線路で接続することによって、PCIシリアルインターフェイスを実現している。LVDS線路を介したシリアル転送は、PCIバスサイクルを伝達するためのデータのみならず、割り込み信号の伝達にも利用される。このように、PCIバスサイクルのためのデータと割り込み信号の双方を同一のLVDS線路を介してシリアル転送することにより、PCIシリアルインターフェイスに必要な信号線数を大幅に低減することができる。

【0024】

PCI-ISAブリッジ16は、PCIバス2とISAバス3とをつなぐブリッジであり、PCIバス2のバスマスターとして動作することができる。ISAバス3上には、各種ISAデバイス18が接続されている。I/Oコントローラ17は、PCIバス2上のバスマスターまたはターゲットとして機能するデバイスである。PCカードコントローラ、IDEコントローラ、サウンドコントローラなどのデバイスが、I/Oコントローラ17としてPCIバス2上に接続される。

【0025】

埋め込みコントローラ(EC)19はPC本体100の電源制御を行うと共に、ドッキングステーション200内に設けられたドッキングステーションコント

ローラ (D S C) 3 6との通信によって、ドッキングステーション2 0 0のドック・アンドドックシーケンスの制御を行う。

【0026】

ドッキングステーション2 0には、図示のように、PCIバス4、ネットワークインターフェイスコントローラ3 1、PCカードコントローラ3 2、IDEコントローラ3 3、PCIスロット3 4、セカンダリPCIシリアル転送コントローラ3 5、ドッキングステーションコントローラ(D S C)3 6、その他のコントローラ3 7等が設けられている。

【0027】

ネットワークインターフェイスコントローラ3 1はLANに接続するための通信制御を行うためのものであり、PCIバス4上のバスマスターまたはターゲットとして機能する。PCカードコントローラ3 2は、PCカードスロットに装着されたPCMCIA/CARDBUS仕様のPCカードの制御を行う。このPCカードコントローラ3 2も、PCIバス4上のバスマスターまたはターゲットとして機能する。IDEコントローラ3 3は、ドッキングステーション2 0内に設けられた例えばハードディスクドライブやCD-ROMドライブなどのIDEデバイスを制御するものであり、PCIバス4上のバスマスターまたはターゲットとして機能する。PCIスロット3 4には、各種PCI拡張カードを装着することができる。

【0028】

PC本体1 0 0をドッキングステーション2 0 0に接続した場合には、これらネットワークインターフェイスコントローラ3 1、PCカードコントローラ3 2、IDEコントローラ3 3、PCIスロット3 4のPCI拡張カードといったPCIデバイスを、PC本体1 0 0内のハードウェアリソースとして使用することが可能となる。

【0029】

(PCI-PCIブリッジ)

図2には、プライマリPCIシリアル転送コントローラ1 5およびセカンダリPCIシリアル転送コントローラ3 5それぞれの機能構成が示されている。

【0030】

前述したように、プライマリPCIシリアル転送コントローラ15およびセカンダリPCIシリアル転送コントローラ35は物理的には独立したLSI同士であるが、論理的には1個のPCI-PCIブリッジとして機能する。よって、プライマリPCIシリアル転送コントローラ15とセカンダリPCIシリアル転送コントローラ35間をつなぐLVDS線路は、PCI-PCIブリッジ内のローカルな内部配線に過ぎず、ソフトウェアからは認識されない。これは、シリアル伝送路に対する無駄なリソースの割り当てが行われないことを意味する。また、プライマリPCIシリアル転送コントローラ15およびセカンダリPCIシリアル転送コントローラ35には、同一のデバイス識別情報が割り当てられており、ソフトウェアからは1個のデバイスとして認識される。もちろん、プライマリPCIシリアル転送コントローラ15およびセカンダリPCIシリアル転送コントローラ35の2つのコントローラが1個のデバイスとして認識されればよいので、デバイス識別情報についてはプライマリPCIシリアル転送コントローラ15のみに持たせてもよい。

【0031】

プライマリPCIシリアル転送コントローラ15およびセカンダリPCIシリアル転送コントローラ35の各々は、PCIインターフェイス部と、シリアル転送インターフェイス部とから構成されている。

【0032】

プライマリPCIシリアル転送コントローラ15においては、PCIインターフェイス部は、プライマリPCIバス2との間でバストランザクションを授受する。一方、セカンダリPCIシリアル転送コントローラ35においては、PCIインターフェイス部は、セカンダリPCIバス4との間でバストランザクションを授受する。PCIインターフェイス部間のトランザクションの授受は、プライマリPCIシリアル転送コントローラ15およびセカンダリPCIシリアル転送コントローラ35にそれぞれ設けられたシリアル転送インターフェイス部間のシリアルデータ転送によって行われる。

【0033】

PCIバスは多ビット幅のアドレス/データ線等を含む並列伝送路であり、PCIバス上のバストランザクションは、基本的には、コマンドおよびアドレス出力のためのアドレスフェーズと、それに後続する1以上のデータ転送フェーズとから構成される。したがって、各PCIインターフェイス部が対応するPCIバスとの間でコマンド、アドレス、データの受け渡しを行い、且つPCIインターフェイス部間のコマンド、アドレス、データの授受をシリアル転送インターフェイス部間のシリアル転送によって行うことによって、プライマリPCIバス2からセカンダリPCIバス4へのトランザクション（PCIバスサイクル）の伝達、およびセカンダリPCIバス4からプライマリPCIバス2へのトランザクション（PCIバスサイクル）の伝達が可能となる。

【0034】

プライマリPCIバス2上のバスマスタからセカンダリPCIバス4上のデバイスへのトランザクションを伝達する場合には、プライマリPCIシリアル転送コントローラ15はプライマリPCIバス2上で実行されるトランザクションのターゲットとなり、セカンダリPCIシリアル転送コントローラ35はセカンダリPCIバス4上で実行されるトランザクションのイニシエータ（バスマスタ）となる。一方、セカンダリPCIバス4上のバスマスタからプライマリPCIバス2上のデバイスへのトランザクションを伝達する場合には、セカンダリPCIシリアル転送コントローラ35はセカンダリPCIバス4上で実行されるトランザクションのターゲットとなり、プライマリPCIシリアル転送コントローラ15はプライマリPCIバス2上で実行されるトランザクションのイニシエータとなる。なお、セカンダリPCIバス4上にバスマスタデバイスが存在しない場合には、前者の場合のみとなる。

【0035】

プライマリPCIシリアル転送コントローラ15およびセカンダリPCIシリアル転送コントローラ35のPCIインターフェイス部には、図示のように、個別にコンフィグレーションレジスタ150, 350が設けられている。コンフィグレーションレジスタ150, 350の各々はPCI仕様に準拠したレジスタ群から構成されており、ここには、同一の環境設定情報が設定される。

【0036】

環境設定情報には、前述のデバイス識別情報を初め、そのデバイスが使用するメモリアドレス空間やI/Oアドレス空間などのハードウェアリソースを指定するためのデバイス制御情報、現在のデバイスの状態を示すデバイスステータス情報などが含まれている。

【0037】

デバイス識別情報はそのデバイスの種類を識別するためのものであり、デバイスID、ベンダID、リビジョンID、ヘッダタイプ、クラスコードなどの情報から構成されている。デバイス識別情報は読み取り専用であり、コンフィグレーションレジスタ150, 350には予め同一のデバイス識別情報が書き込まれている。もちろん、この読み取り専用のデバイス識別情報については、CPU11に近い側のプライマリPCIシリアルコントローラ15側にのみ用意しておけばよく、セカンダリPCIシリアルコントローラ35側に設ける必要はない。CPU11によるコンフィグレーションサイクルでアクセスされるのはプライマリPCIシリアルコントローラ15側だけであり、そのプライマリPCIシリアルコントローラ15からのデバイス識別情報の読み取りによって、PCIバス2と4との間にPCI-PCIブリッジが存在することが認識されるからである。

【0038】

コンフィグレーションレジスタはPCIデバイスの動作環境を規定するための環境設定情報を保持するためのものであり、PCIデバイスには必ず1個のコンフィグレーションレジスタが設けられる。プライマリPCIシリアル転送コントローラ15およびセカンダリPCIシリアル転送コントローラ35は1個のPCIデバイス(PCI-PCIブリッジ)として動作するものであるので、基本的には、前述したように、1個のコンフィグレーションレジスタをコントローラ15, 35間で共用することもできる。しかし、このように一方のコントローラにのみコンフィグレーションレジスタを設けた場合には、コンフィグレーションレジスタが設けられていない方のコントローラについては、トランザクション処理を行う度にシリアル伝送路を介して相手のコントローラのコンフィグレーションレジスタをリードしなければならず、それによってシステム性能の低下が招かれ

る。本実施形態のようにコンフィグレーションレジスタを2つのコントローラ15, 35に個別に実装することにより、2つのコントローラ14, 35は、それぞれ自身のコンフィグレーションレジスタに設定されている環境設定情報にしたがって動作することができるので、高速動作が可能となる。この場合、2つのコントローラ14, 35のそれぞれに設けるのは、読み取り専用のデバイス識別情報を設定するためのレジスタを除く、他のほとんどのレジスタ群（デバイス制御情報を設定するためのレジスタ群、現在のデバイスの状態を示すデバイスステータス情報を設定するためのレジスタ群等）である。

【0039】

コンフィグレーションレジスタ150と350の内容の同一性は、次のようなコピー動作をプライマリPCIシリアル転送コントローラ15とセカンダリPCIシリアル転送コントローラ35との間で自動実行することによって実現される。すなわち、CPU11がコンフィグレーションレジスタ150にデータを書き込むためのライトトランザクション（コンフィグレーションライトサイクル）を実行した場合には、コンフィグレーションレジスタ150からコンフィグレーションレジスタ350に対するコピー動作が自動的に行われ、その後に、CPU11に対して書き込み完了を示すステータスが返される。これにより、常に、コンフィグレーションレジスタ150と350の内容を同一に保つことができる。

【0040】

（2つのPCIシリアル転送コントローラの内部構造）

次に、図3を参照して、プライマリPCIシリアル転送コントローラ15およびセカンダリPCIシリアル転送コントローラ35それぞれの内部構造を、論理的なプロトコル階層構造に着目して説明する。

【0041】

プライマリPCIシリアル転送コントローラ15は、図示のように、トランザクションバッファ201、バスサイクルコントローラ202、ブロック転送バッファ203、ワードバッファ204、シリアル・パラレルコンバータ205、およびLVDS送受信部206等から構成されている。

【0042】

トランザクションバッファ201およびバスサイクルコントローラ202は前述のPCIインターフェイス部に相当し、またブロック転送バッファ203、ワードバッファ204、シリアル・パラレルコンバータ205、およびLVDS送受信部206が前述のシリアルインターフェイス部に相当する。

【0043】

同様に、セカンダリPCIシリアル転送コントローラ35は、図示のように、トランザクションバッファ301、バスサイクルコントローラ302、ブロック転送バッファ303、ワードバッファ304、シリアル・パラレルコンバータ305、およびLVDS送受信部306等から構成されている。トランザクションバッファ301およびバスサイクルコントローラ302は前述のPCIインターフェイス部に相当し、またブロック転送バッファ303、ワードバッファ304、シリアル・パラレルコンバータ305、およびLVDS送受信部306が前述のシリアルインターフェイス部に相当する。

【0044】

図3の右端には、本実施形態のPCIシリアルインターフェイスを実現するためのプロトコル階層構造が示されている。最上位階層は、PCIバストランザクションレイヤであり、その下が、実際にトランザクションを実行するために必要なバスサイクルを制御するためのPCIバスサイクルレイヤである。

【0045】

バスサイクルレイヤから上の部分を、左右の半分を合わせて一体として実現したもの、つまり、バストランザクションバッファ202、301、およびバスサイクルコントローラ202、302を合わせたものが、通常のPCI-PCIブリッジに相当するものとなる。

【0046】

図3の下半分のレイヤが、プライマリPCIシリアル転送コントローラ15とセカンダリPCIシリアル転送コントローラ35との間でシリアル通信を行うための部分である。

【0047】

上半分がPCIバスのプロトコルに従って設計されるのに対し、下半分は、P

PCIバス上を転送されるデータを忠実に相手方へ送るのに最適となるように設計される。転送するデータがPCIバスの上でどんな意味をもつかを考慮する必要はなく、そのデータがPCIバス上で持つ意味に適した転送特性を正しくアサインし、実現すればよい。通信の世界でのパケット通信に近い考え方をとることができる。

【0048】

その意味では、図中のワード(WORD)が固定長のパケットに当たり、ブロック(BLOCK)は、1WORDの制御ワードと、0から10WORDのデータワードとを含む転送単位である。

【0049】

トランザクションバッファ201, 301は、PCIバスサイクルをトランザクションとして管理するためのバッファであり、PCIバスサイクルと後述のブロック転送との仲立ちに使用される。トランザクションを構成する情報は、トランザクションの種類によって多少異なるが、

- ・アドレス
- ・コマンド
- ・ライトデータ(ライト系トランザクション)
- ・バイトイネーブル
- ・完了ステータス
- ・リードデータ(リード系トランザクション)

などである。これら情報はトランザクションバッファ201, 301に保持される。

【0050】

ブロック転送バッファ(BLOCK)204, 304は、ブロックと称する可変長のデータを、2つのシリアルコントローラ15, 35間でまとめて転送するための一時的な情報格納場所である。ブロックサイズは前述したように可変長であり、基本的には、ある1つのトランザクションを構成する、アドレス、データ、コマンド、バイトイネーブルなどの情報から構成される。

【0051】

ワードバッファ (WORD) 204, 304は、ワード (WORD) と称される固定長のデータを、2つのシリアルコントローラ 15, 35間でひとつづつ転送するための一時的な情報格納場所である。ワードには制御ワードとデータワードとがある。ブロックの実体部分 (PCI トランザクション情報: アドレス、コマンド、データ、バイトイネーブルなど) はデータワードとして受け渡され、それ以外の各種制御情報は制御ワードとして受け渡される。

【0052】

シリアル・パラレルコンバータ 205, 305は、ワード単位でのパラレル/シリアル変換、およびシリアル/パラレル変換を行う。LVD�送受信部 206, 306は、LVD�線路を介した実際のシリアルデータ転送を行う。

【0053】

(バッファ構造)

次に、図4を参照して、具体的なバッファ構造について説明する。

【0054】

ブロックバッファには、取り扱うPCIトランザクションの違いによって、以下の2種類ある。

- ・ Expressバッファ (ポステッドメモリライト系のトランザクション用)
- ・ BLOCKバッファ (それ以外のトランザクション用)

また、データの流れの方向により、以下の2種類がある。

- ・ Outgoingバッファ (送信するブロックの情報を格納する)
- ・ Incomingバッファ (受信したブロックの情報を格納する)

これらの組み合わせて、以下の合計4種類のバッファがある。

- ・ Outgoing Expressバッファ (OEB)
- ・ Outgoing BLOCKバッファ (OBB)
- ・ Incoming Expressバッファ (IEB)
- ・ Incoming BLOCKバッファ (IBB)

本実施形態では、システムとしての性能を確保するため、OBB, OEB, IBB, IEBとも、それぞれ4段ずつ用意している。その意味で、4種類のブロックバッファ FIFO、つまりOBB_FIFO, OEB_FIFO, IBB_FIFO

FIFO, IEB_FIFOが設けられることになる。

【0055】

即ち、図4に示すように、ブロックバッファ203には、4段のOutgoing BLOCKバッファ(OBB)203aから構成されるOBB_FIFOと、4段のOutgoing Expressバッファ(OEB)203bから構成されるOEB_FIFOと、4段のIncoming BLOCKバッファ(IBB)203cと、4段のIncoming Expressバッファ(IEB)203dが設けられている。同様に、ブロックバッファ303にも、4段のOutgoing BLOCKバッファ(OBB)303aから構成されるOBB_FIFOと、4段のOutgoing Expressバッファ(OEB)303bから構成されるOEB_FIFOと、4段のIncoming BLOCKバッファ(IBB)303cと、4段のIncoming Expressバッファ(IEB)303dが設けられている。

【0056】

前述したように、ブロックとは、構造を持ったワード(WORD)のまとまりである。ブロックは次の3つのパートから構成される。送信も、受信も、時間的に以下の順序で処理される。

【0057】

- ・一つの制御ワード
- ・複数のデータワード(ゼロ以上10以下のデータワード)
- ・一つのチェックサムワード

ワードとは、bitのまとまりである。大きく分類すると、制御情報を担う制御ワードと、データを担うデータワードとに分けられる。ブロックバッファ203, 203では、ブロック構成する各ワードを以下のような17ビットを単位として取り扱う。

【0058】

- ・ブロックを構成する各ワードの情報の実体サイズが16ビット
- ・制御ワードとデータワードを区別するためのフィールドが1ビット

ブロックバッファ203, 303は、下位の処理階層との間で、17ビットのワードを単位としてデータの受け渡しを行う。下位の処理階層は、前述のワードバッファと、ビットレイヤである。ビットレイヤは図3のシリアルパラレルコン

バータ205, 305と、LVDS送受信部206, 306に相当するものである。以下の処理は、下位の処理階層で行われる処理であり、ブロックバッファ203, 303は閲知しない。

【0059】

・ワードバッファ： ブロックバッファが扱う各17ビットのワードに対して1bitのパリティが付加され、18bitのデータが作られる。

・ビット階層： 18bitのデータがパラレルデータからシリアルデータに変換されて、高速シリアル転送される。実際には、送信用の2本のLVDS線と、受信用の2本のLVDS線が各コントローラ15, 35に設けられているので、18bitのデータが9bit×2つのストリームに分解されて、それがパラレルデータからシリアルデータに変換された後に同時にシリアル転送される。

【0060】

・受信側はこの逆のプロセスとなる。

【0061】

(ブロック転送)

前述したように、本実施形態では、下位の階層ではシリアル転送が行われるが、ブロックバッファ203と303との間ではブロックを単位としたデータ転送(ブロック転送)が実行される。このブロック転送は、トランザクションバッファ201, 301の要求にサービスするために行われるものであり、トランザクションバッファから渡された情報を、相手側PCIシリアル転送コントローラのトランザクションバッファまで、正確に、且つ速やかに送り届けることが目的である。この目的を実現するため、ブロックバッファ階層では、送信対象のブロックに以下の情報を付加する。

【0062】

・ブロックを識別するためのブロックID(制御ワードに埋め込んで送信する)

・チェックサムワード(制御ワードの一種)

送信対象のブロックデータそれそれには、連続した値のブロックIDが付加さ

れる。理想的には零～無限大までの値をブロックIDとして順番に割り当てられればよいが、実際には、使用できるbit数には限界があるため、本実施形態では、3bitのブロックIDを用意し、0～7までを、この順で繰り返し割り当てるにすることにする。

【0063】

(2つのPCIシリアル転送コントローラの内部構造)

次に、図5を参照して、プライマリPCIシリアル転送コントローラ15およびセカンダリPCIシリアル転送コントローラ35それぞれの内部構成を説明する。

【0064】

プライマリPCIシリアル転送コントローラ15およびセカンダリPCIシリアル転送コントローラ35の構成は基本的に同一である。これら各コントローラは、図示のように、PCIバスコントロールブロック(PCI_CTL)401、PCIバスアービトリエーションブロック(ARBIT)402、サイクルデコードブロック(CYCDEC)403、コンフィグレーションレジスタブロック(CF_REG)404、トランザクションバッファ&コントロールブロック(TBC)405、ブロック&ワードバッファブロック(BWB)406、ブロック転送バッファブロック(BB)407、ワードバッファブロック(WB)408、ビットレイヤーブロック(BLB)409、Misc情報更新ブロック(MIS)410、およびシリアルインターラプト同期ブロック(SIS)411から構成されている。

【0065】

PCIコントロールブロック(PCI_CTL)401は、PCIバスマスターおよびターゲットとして、PCIバスインターフェースを制御するためのものであり、マスタレイテンシティマ(MLT)を備えている。マスタレイテンシティマ(MLT)は、現在のサイクルを実行しているバスマスターに対してサイクルを中止させるためのタイミングを計時するためのものである。ターゲット動作時には、PCIバスコントロールブロック(PCI_CTL)401は、サイクルデコードブロック(CYCDEC)403からのヒット信号をトリガとして動作する

。すなわち、PCIバスコントロールブロック（PCI_CTL）401は、ヒット信号を受けると、PCIバス上の現在のバスマスターによって行われるPCIバスサイクルに応答して、ターゲットとしてのPCIバスサイクルを開始する。

【0066】

PCIバスアービトリエーションブロック（ARBIT）402は、PCIバス上でPCIバスマスターのアービトリエーションを行うためのアービタである。

【0067】

サイクルデコードブロック（CYCDEC）403は、PCIバスターゲット動作時、PCIサイクル種別およびアドレスをデコードしてヒット判定を行う。また、このサイクルデコードブロック（CYCDEC）403は、トランザクションバッファ&コントロールブロック（TBC）405に対するサイクルスタート信号およびPCIバスコントロールブロック（PCI_CTL）401に対するヒット信号の生成を行う。

【0068】

コンフィグレーションレジスタブロック（CF_REG）404は、前述のPCIコンフィグレーションレジスタである。

【0069】

トランザクションバッファ&コントロールブロック（TBC）405はPCIバスサイクルをトランザクションとして管理するバッファであり、ブロック転送バッファブロック（BB）407とPCIバスコントロールブロック（PCI_CTL）401との間でのデータ受け渡しの制御を行う。

【0070】

ブロック&ワードバッファブロック（BWB）406は、前述のブロックバッファとその制御ロジックから構成されるブロック転送バッファブロック（BB）407と、前述のワードバッファとその制御ロジックから構成されるワードバッファブロック（WB）408との2つのブロックを便宜上1つにまとめたものである。ブロック転送バッファブロック（BB）407は、トランザクションバッファ&コントロールブロック（TBC）405またはワードバッファブロック（WB）408との間で授受されるデータを一時的に格納するために用いられる。

また、バッファとして、ポストライト用 (OEB/IEB) と、ポストライト以外のトランザクション用 (OBB/IBB) をそれぞれ独立にもつ。図中のOEB/OBBが送信用バッファ、IEB/IBBが受信用バッファである。この送信用および受信用バッファは、それぞれピットレイヤーブロック409内のPLLで生成される非同期の送信用および受信用クロックで動作する。PCIクロックを使用していないので、ブロック転送バッファブロック(BB)407より上位層のブロックとは非同期となる。また、このブロック&ワードバッファブロック(BWB)406は、チェックサム生成、シリアルバス上でのエラーチェックおよび再送処理の制御等も行う。

【0071】

一方、ワードバッファブロック(WB)408は、ブロック転送バッファブロック(BB)407からの固定長のデータ(BLOCK)をピットレイヤーブロック(BLB)409との間でWORD単位で1つずつ転送するために一時的に格納するためのバッファである。図中のOWBは送信用バッファ、IWBは受信用バッファであり、それぞれピットレイヤーブロック(BLB)409内のPLLで生成される非同期の送信用および受信用クロックで動作する。

【0072】

ピットレイヤーブロック(BLB)409は、ワードバッファブロック(WB)408からの固定長のデータ(WORD)を2つに分け、2系統のシリアルデータとしてシリアル通信を行うものであり、送信側は、パラレル→シリアル変換を行い、受信側は、シリアル→パラレル変換を行う。また、このピットレイヤーブロック(BLB)409は、送信用、受信用にそれぞれPLLを持ち、送信用PLLは、所定のクロック(PLCLK)入力をもとに動作(9倍)し、受信用PLLは、セカンダリPCIシリアル転送コントローラ35からLVDS線路を介して送信されるLVDSシリアル受信クロック(LVDC_I)をもとに動作する。なお、LVDSシリアル送信クロック(LVDC_O)は、PLCLK入力と同じ周波数の出力となる。

【0073】

Misc情報更新ブロック(MIS)410は、PCIバスの割り込み信号(

INT [A:D] #) をMisc情報として扱い処理するためのブロックである。シリアルインターラプト同期ブロック (SIS) 411は、ISAデバイスなどからのレガシイ割り込み信号を処理するためのブロックである。

【0074】

(フロー制御)

次に、送信側コントローラのOBB/OEBと、受信側コントローラのIBB/IEBとの間の通信の流れを制御するためのフロー制御について、その原理を説明する。

【0075】

前述したように、ブロックバッファ階層の役目・期待されていることは、トランザクションバッファから委託された情報を、正確に、且つ速やかに、受け渡すことである。これを実現するため、本実施形態のフロー制御方法では、以下の2つの方針の双方が組み合わせて用いられる。

【0076】

(1) “正確に”の実現

・送信側コントローラのOBB/OEBは、トランザクションバッファから受け取った情報についての責任を負う。つまり、単に送信を行えば“終わり”になれるのではない。受信側コントローラの対応するIBB/IEBへ確実に伝えたことを確信できるまでは、OBB/OEBは勝手に“終わり”になってはならない。また、必要ならば、再度、送信することも行わなくてはならない。

【0077】

・受信側コントローラのIBB/IEBは、ブロックを正しく受信したことを、送信側コントローラの送り元のOBB/OEBへ通知（返信）する義務を負う。受信したブロックをトランザクションバッファに引き渡せば終わりではなく、OBB/OEBへの通知が必要である。

【0078】

・その通知は、肯定応答ACK、具体的にはACKB/ACKEという特別な制御ワードにより行われる。

・ACKB/ACKEは、IBB/IEBからOBB/OEBへ向けて、発

行されるものである。各ACKB/ACKEには、受信したブロックデータと同じブロックIDが付加される。ACKB/ACKEは、一つのブロックデータに対してただ一回しか発行されない。

【0079】

(2) “速やかに”の実現

- 送信側コントローラのOBB_FIFO/OEB_FIFOは、投機的なブロックの先出し送信をすることができる。つまり、まだ、以前に送信したOBB/OEBに対してACKB/ACKEが返信されていなくとも、別の送信待ちのOBB/OEBがあれば、そのOBB/OEBからのブロックデータの送信を開始することができる。すなわち、ブロックデータの送信は、ACKB/ACKEの返信を待たずに、順次実行される。

【0080】

- 到着したブロックを例えばバッファオーバーなどによって受信できないとき、受信側コントローラのIBB_FIFO/IEB_FIFOは、そのことを、送信側コントローラのOBB_FIFO/OEB_FIFOへ通知する。

【0081】

- その通知は、否定応答NACK、具体的にはNACKB/NACKEという特別な制御ワードにより行われる。

- NACKB/NACKEは、IBB_FIFO/IEB_FIFOからOBB_FIFO/OEB_FIFOへ向けて、再送要求として発行される。

【0082】

- NACKB/NACKEを受け取ったOBB_FIFO/OEB_FIFOは、投機的先出しが失敗したことを認識し、適当な再送信の処理を行う。どのような再送信を行うかは、IBB_FIFO/IEB_FIFOとOBB_FIFO/OEB_FIFOとの間の取り決めによって決定されるが、基本的には、送信済みのブロックデータの内、ACKB/ACKE待ちのブロックデータの再送信処理がブロックID順に行われる。

【0083】

(OBBのステートマシン)

次に、図6を参照して、本実施形態で使用されるOBBの状態遷移について説明する。

【0084】

前述したように、OBBは4個（4段）ある。ここでは、そのうち1個（1段）に着目し、そのステートマシンについて説明する。なお、OEBのステートマシンもOBBと同様である。OBBの基本的な状態は次の通りである。

- ・ “初期状態”：送信すべきブロックを持っていない状態
- ・ “送信待ち状態”：トランザクションバッファからブロックデータを委託され、送信できる順番が来るのを待っている状態
- ・ “返信到着待ち状態”：ブロックデータの送信を終えたが、ACKBがまだ返信されて来ない状態

状態遷移は、“初期状態” → “送信待ち状態” → “返信到着待ち状態” → “初期状態” → … のように繰り返すことになる。送信済みのデータであっても、ACKを受けるまではそのままバッファ内に保持される。

【0085】

具体的には、OBBには、図6に示すような5つの状態が定義される。

<NULL>

送信すべきブロックの実体を持っていない。前述の“初期状態”に相当する。

<TX_REQ>

トランザクションバッファから、送信すべきブロックの実体を受け取ったが、送信は行っていない。あるいは、一度送信したが再送要求を受け、送信できるようになるまで待っている状態である。前述の“送信待ち状態”的一種である。

<TX_OUT>

ブロックの送信中の状態であり、前述の“送信待ち状態”的一種である。

<ACK_WAIT>

一つのブロックを最後まで送信し終え、ACKBが到着するのを待っている状態である。前述の“返信到着待ち状態”に相当する。

<TX_SYNC>

エラー回復処理の完了を待っている状態である。エラー回復の結果により、“初期状態”あるいは“送信待ち状態”に遷移する。

【0086】

なお、エラー回復処理は、2つのコントローラ15, 35間の通信路に何らかのエラーが発生した場合に実行される処理である。

エラーとは、通常のフロー制御プロトコルの範囲では処理できない事象の発生である。静電気ノイズのような外部要因による通信信頼性の低下が、エラーの原因である。エラーを検出すると、ブロックバッファ階層は、通常の動作を中断し、エラー回復に専念する。一回のエラー回復処理に要する時間は、10マイクロ秒から100マイクロ秒である。これは、受信側PLLの周波数同期・位相同期の取り直しに、この程度の時間がかかるからである。

【0087】

エラーを検出する階層は、ブロックバッファ階層を含む、それ以下の階層である。トランザクションバッファに対しては、エラー発生は隠蔽される。この間、PCIバス側からは、バス上でウェイトがかかったかのように見える。

【0088】

(フロー制御モデル)

図7には、ブロックバッファ203, 303間で行われるフロー制御のモデルが示されている。ブロックバッファ203側には、ブロック転送プロトコル処理部501とエラー回復処理プロトコル処理部502が設けられる。同様に、ブロックバッファ303側にも、同じ構成のブロック転送プロトコル処理部501とエラー回復処理プロトコル処理部502が設けられる。ブロック転送プロトコル処理部501は、再送処理を含む、フロー制御プロトコル処理を実行する。一方、エラー回復処理プロトコル処理部502は、エラー回復処理を実行する。このように2種類のプロトコル処理部501, 502を用意することにより、フロー制御プロトコル処理における再送制御とエラー回復処理とを切り離すことができる。

【0089】

エラー回復処理では、初期化処理時と同様に、コントローラ15, 35間の通

信路の同期確立のためのピットパターン転送から始められる。そして、コントローラ間の論理的な不一致の解消を行うために、受信側から送信側に、受信済みのブロックデータのブロックIDが通知された後、通知された受信済みのブロックデータの次のブロックデータからブロックデータが再開される。

【0090】

エラー回復処理プロトコル処理部502を起動されるエラー検出条件の代表的なものは次の通りである。

【0091】

- ・チェックサムエラー（受信側のブロックバッファ階層で検出する）
- ・ACKB／ACKEのブロックIDの不連続（送信側のブロックバッファ階層で検出する）
- ・パリティエラー検出（受信側のブロックバッファ階層で検出する）

次の条件はエラーではない。

【0092】

- ・OBB_FIFO／OEB_FIFOがNACKB／NACKEを受信（正常なフロー制御プロトコルの範囲内で起こる事象であり、この場合には再送制御が実行される）
- ・IBB_FIFO／IEB_FIFO側へ到着するブロックのブロックIDの不連続（NACKB／NACKEに対する再送制御により、正常なフロー制御プロトコルの範囲内で起こる事象ある）
(フロー制御のための機能構成)

図8には、フロー制御プロトコル処理のために必要な送信側および受信側それぞれの機能構成が示されている。

【0093】

図示のように、送信側には、ブロック送信部501、ACK待ちブロック管理部502、再送制御部503、およびエラー検出部504が設けられている。受信側には、ブロック受信部601、ACK送信部602、NACK送信部603、およびエラー検出部604が設けられている。実際には、図7のブロック転送プロトコル処理部501には、これら送信側と受信側の双方の機能が設けられて

いる。

【0094】

ブロック送信部501は、送信対象の複数のブロックデータそれぞれに連続するブロックIDを付加し、それら複数のブロックデータ(BLK)を、受信側からのACKE/ACKBの返信を待たずに、送信側のコントローラから受信側のコントローラに順次送信する。ACK待ちブロック管理部502は、前述のOB Bステートマシンで説明した“返信到着待ち状態”のブロック(以下、ACK待ちブロックと称する)を管理するためのものであり、送信済みのブロックデータ毎に受信側のコントローラからのACKE/ACKBの返信の有無を検出する。再送制御部503は、受信側のコントローラからNACKE/NACKBによる再送要求を受けた場合、ACK待ちブロックの再送をブロック送信部501に実行させる。エラー検出部504は、ACKE/ACKBのブロックIDの不連続の有無により、エラー発生の検出を行う。エラー発生が検出されると、図7のエラー回復プロトコル処理部502が起動される。

【0095】

ブロックIDの不連続には、到着したACKE/ACKBのブロックIDが連続しない場合のみならず、到着したACKE/ACKBのブロックIDがACKE/ACKB待ちのブロックとして存在しない場合も含まれる。

【0096】

ブロック受信部601は、ブロック送信部501からのブロックデータの受信を行い、IBB/IEBに格納する。ACK送信部602は、ブロックデータが正常に受信される度に、その受信したブロックデータのブロックIDと同一のブロックIDが付加されたACKE/ACKBを、受信側のコントローラから送信側のコントローラに送信する。NACK送信部603は、送信側コントローラから送信されるブロックデータを受信側コントローラで受信することができないとき、あるいは受信したブロックデータのブロックIDが不連続であるときなどに、NACKE/NACKBを送信側のコントローラに送信する。エラー検出部604は、パリティーエラーやチェックサムエラーの有無により、エラー発生の検出を行う。エラー発生が検出されると、図7のエラー回復プロトコル処理部50

2が起動される。

【0097】

(ブロック転送&再送制御)

次に、図9を参照して、プライマリPCIシリアル転送コントローラ15とセカンダリPCIシリアル転送コントローラ35との間で実行されるフロー制御動作（ブロック転送&再送制御）について説明する。

【0098】

ここでは、プライマリPCIシリアル転送コントローラ15を送信側コントローラ、セカンダリPCIシリアル転送コントローラ35を受信側コントローラとする。

【0099】

送信側コントローラでは、送信待ちの複数のブロックデータにそれぞれ連続するブロックIDが付加され、それら複数のブロックの投機的な先出し送信が順次行われる。図9においては、ブロックBLK#0からブロックBLK#3までの4つのブロックの先出し送信を行う場合が示されている。4つのブロックの先出し送信後は、受信側コントローラからのACKの返信の度に次のブロックの送信が行われる。これにより、ACK待ちのブロックは最大4個となり、3bitのブロックでもコントローラ間の論理的な不一致の発生を防止することができる。

【0100】

最初のブロックBLK#0が受信側コントローラにて正常に受信されると、ブロックID=#0のACK(ACK#0)が返信される。ACK#0が到着すると、送信側コントローラでは、ブロックBLK#0を保持していたOBB/OEBはACK待ちの状態から初期状態に遷移し、ACK待ちのブロックデータはブロックBLK#1～ブロックBLK#3となる。そして、次の送信待ちのブロックBLK#4の送信が開始される。

【0101】

ブロックBLK#1が受信側コントローラにて正常に受信されると、ブロックID=#1のACK(ACK#1)が返信される。ACK#1が到着すると、送信側コントローラでは、ブロックBLK#1を保持していたOBB/OEBはA

CK待ちの状態から初期状態に遷移し、ACK待ちのブロックデータはブロックBLK#2～ブロックBLK#4となる。そして、次の送信待ちのブロックBLK#4の送信が開始される。

【0102】

なお、実際には、ブロックBLK#0～#3の全ての先出し処理が終了する前に、ACK#0、あるいはACK#0およびACK#1の返信が行われるので、ブロックBLK#4、BLK#5の送信は、先出し処理の後、連続的に行われる。

【0103】

受信側コントローラのIBB/IEBバッファのオーバーフローや、ブロックBLK#5までのブロックIDに不連続が生じた場合、受信側コントローラはブロックBLK#5を受信することができない。この場合、NACK(NACKB/NACKE)が送信側コントローラに送信される。

【0104】

NACKが到着すると、送信側コントローラでは、ACK待ちのブロック、つまり“返信到着待ち状態”的OBB/OEBを検出する。本例では、ACK#0とACK#1が既に到着しているので、ACK待ちのブロックは、ブロックBLK#2、BLK#3、BLK#4、BLK#5となる。そして、送信側コントローラは、ACK待ちブロックの再送を開始する。この再送処理は、ACK待ちの先頭ブロックBLK#2からの先出し送信処理によって実行される。具体的には、次の手順が実行されることになる。

【0105】

(1) OBB_FIFO/OEB_FIFOは、NACKB/NACKEを受け取ると、送信途中(仕掛けり中)のブロックがあれば、それだけは、送信してしまう。

【0106】

(2) ACKB/ACKE待ちになっているすべてのOBB/OEBの状態を、送信待ちに引き下げる。そのうえで、送信待ちになっているOBB/OEBを、あらためて順番に送信する。

【0107】

そして、この再送処理後、ブロックBLK#2が受信側コントローラで受信され、ACK#2が返信されると、今度は、次の送信待ちのブロックBLK#6の送信が開始される。

【0108】

このように、本実施形態のフロー制御においては、送信側から受信側へはブロックデータの送信、および受信側から送信側へのACKの返信のどちらにおいてもブロックIDを附加しているので、ブロックデータの送信とACKの返信を非同期で行うことが可能となる。この仕組みにより、送信側から受信側へのブロックデータの先出し送信が実現されている。さらに、送信側では、受信側がどのブロックデータの受信まで完了しているかを知ることができるので、例えば受信側のバッファのオーバーフローなどによって受信側がブロックデータを受信できなかつた場合には、未受信のブロックデータを正しく再送することができる。

【0109】

なお、NACKB/NACKEを受け取ったときに、とりあえず、送信待ちのOBB/OEBをすべて送信してしまい、その後で、あらためて最初から再送信を始める、という方法を使用することも可能である。

【0110】

(ブロック転送&再送制御 その2)

次に、図10を参照して、プライマリPCIシリアル転送コントローラ15とセカンダリPCIシリアル転送コントローラ35との間で実行されるフロー制御動作(ブロック転送&再送制御)の第2の例について説明する。

【0111】

本例は、NACKB/NACKEとは無関係にブロックを繰り返し送信する方式である。すなわち、OBB_FIFO/OEB_FIFOは、送信待ちのOBB/OEBだけでなく、ACKB/ACKE待ちのOBB/OEBをも、順番に、繰り返し送信を行う。IBB_FIFO/IEB_FIFO側では、NACKB/NACKEを発行する必要がない。これにより、制御は極めて簡単になる。

【0112】

IBB_FIFO/IEB_FIFOは、自分の都合がいいときに、到着したブロックをIBB/IEBに受信させる。IBB/IEBは、ブロックを正しく受信すると、そのことをブロックID付きのACKB/ACKEの発行により、OBB/OEBに通知する。

【0113】

図10の例では、前述の図9の場合と同様に、最初に、ブロックBLK#0からブロックBLK#3までの4つのブロックの先出し送信が行われる。これらブロックにはそれぞれ連続したブロックIDが付加されている。4つのブロックの先出し送信後は、それら送信済みのブロックBLK#0～ブロックBLK#3はACK待ちのブロックとなる。受信側コントローラからのACKの返信があるまでは、送信側のコントローラは、ACK待ちのブロックBLK#0～ブロックBLK#3の連続送信を繰り返し実行する。

【0114】

最初のブロックBLK#0が受信側コントローラにて正常に受信されると、ブロックID=#0のACK(ACK#0)が返信される。ACK#0が到着すると、送信側コントローラでは、ブロックBLK#0を保持していたOBB/OEBはACK待ちの状態から初期状態に遷移し、ACK待ちのブロックデータはブロックBLK#1～ブロックBLK#3となる。そして、次の送信待ちのブロックBLK#4と、ACK待ちの残りの3つのブロックBLK#1～ブロックBLK#3から構成される4つのブロックを、連続的に先出し送信する。受信側コントローラから次のACKが返されるまで、BLK#4, BLK#1, BLK#2, BLK#3の連続送信は繰り返し実行される。

【0115】

受信側コントローラは、任意のタイミングでBLK#2, BLK#3, BLK#4をその順で受信すれば良い。よって、NACKB/NACKEによる再送要求は発行されない。

【0116】

ブロックBLK#1が受信側コントローラにて正常に受信され、ブロックID=#1のACK(ACK#1)が返信されると、今度は、BLK#5, BLK#

4, BLK#1, BLK#2の連續送信が繰り返し行われることになる。このように、ACKを受ける度に、新たな送信待ちのブロックが加えられるように連續送信される4個のブロックの組み合せを更新しながら、4個のブロック単位での連續送信が繰り返し行われる。

【0117】

このような繰り返し送信方式は、図8のブロック送信部501が、送信待ちブロックだけでなく、ACK待ちブロックをも加えて、予め決められた複数個単位での先出し送信を繰り返し行うようにするだけで容易に実現することができる。また、再送制御部503およびNACK送信部603は不要となる。

【0118】

(エラー回復処理)

次に、図11を参照して、プライマリPCIシリアル転送コントローラ15とセカンダリPCIシリアル転送コントローラ35との間のブロック転送でエラーが発生した場合に実行されるエラー回復処理について説明する。

【0119】

ここでは、図9または図10の手順でブロック転送を行っている途中で、受信側コントローラからのACKB/ACKEが消失した場合を想定する。すなわち、図示のように、受信側コントローラはBLK#0, BLK#1, BLK#2をそれぞれ受信する度に、ACK#0, ACK#1, ACK#2を返信するが、ACK#1が何らかの原因で途中で消失してしまった場合、送信側コントローラには、ACK#0の後にACK#2が到着してしまう。このACKの不連続により、送信側コントローラはエラーの発生を検出し、通常のフロー制御動作を中断してエラー回復プロトコル処理に移行する。

【0120】

エラーを検出した側のコントローラ（ここでは送信側コントローラ）からの指示により、他方のコントローラ（ここでは受信側コントローラ）もエラー回復プロトコル処理に移行する。エラー回復プロトコル処理は、2つのフェーズから構成される。物理的同期回復フェーズと論理的同期回復フェーズである。物理的同期回復フェーズでは、予め決められた同期用ビットパターンの授受が2つのコン

トローラ間で何度も繰り返し実行される。同期用ビットパターンの授受は専用の制御ワードによって行われる。2つのコントローラが共に物理的同期が確立した状態になり、PLL同期等を始め、2つのコントローラ間のシリアルデータ転送が正常に行うことが可能な条件が揃うと、受信側コントローラからは、どのブロックまで正常に受信する事ができたかを示す情報が専用の制御ワードによって通知される。これにより、受信側コントローラがACKを返信したブロックと、送信側コントローラがACK到着を確認したブロックとが論理的に一致する。そして、互いにエラー回復状態になったことを通知し合うための情報交換を行った後、送信側コントローラおよび受信側コントローラは共に通常のフロー制御プロトコルに戻る。

【0121】

送信側コントローラは、受信側コントローラが通知された受信完了ブロックBLK#2の次のブロックBLK#3から、送信を開始する。

【0122】

以上のように、再送制御とエラー回復制御とを用いることにより、ブロック消失、ACK消失、NACK消失に対しても、正常な状態への回復が可能となる。

【0123】

なお、本実施形態では、PC本体100とドッキングステーション200にそれぞれ、ブリッジを構成する2つのコントローラを分割配置したが、例えばPC本体100に第1の拡張ユニットを接続し、その第1の拡張ユニットを介してさらに第2の拡張ユニットを接続するような場合には、第2の拡張ユニットから見れば、コンピュータ本体と第1の拡張ユニットがホスト装置として機能することになる。この場合、第1の拡張ユニットと第2の拡張ユニットにそれぞれプライマリPCIシリアル転送コントローラ15とセカンダリPCIシリアル転送コントローラ35を分散して設けてもよい。

【0124】

また、ブロック毎にIDを付して管理したが、この場合のブロックの単位は任意のものでよい。

【0125】

【発明の効果】

以上説明したように、本発明によれば、バス間でトランザクションを伝達するために必要なデータを2つのコントローラ間で正確、且つ迅速に転送できるようになり、十分なスループットと動作の信頼性の向上を図ることが可能となる。特に、論理的には1つのバスブリッジとして動作する物理的に異なる2つのコントローラ間のシリアル転送に適用することにより、バスブリッジとしての動作の信頼性および性能を維持した状態で、ホスト装置と拡張ユニット間のシリアル接続を実現することができる。

【図面の簡単な説明】

【図1】

本発明の一実施形態に係るコンピュータシステムの構成を示すブロック図。

【図2】

同実施形態で使用されるPCI-PCIブリッジの構成を示すブロック図。

【図3】

同実施形態で使用されるプライマリPCIシリアル転送コントローラおよびセカンダリPCIシリアル転送コントローラそれぞれの内部構造を示すブロック図。

【図4】

同実施形態で使用されるプライマリPCIシリアル転送コントローラおよびセカンダリPCIシリアル転送コントローラそれぞれにおけるバッファ構造を示す図。

【図5】

同実施形態で使用されるプライマリPCIシリアル転送コントローラおよびセカンダリPCIシリアル転送コントローラそれぞれの具体的なハードウェア構成を示すブロック図。

【図6】

同実施形態で使用されるOBBの状態遷移を説明するための図。

【図7】

同実施形態のプライマリPCIシリアル転送コントローラとセカンダリPCI

シリアル転送コントローラとの間で実行されるフロー制御の原理を説明するための図。

【図8】

同実施形態のフロー制御を実現するための機能構成を示すブロック図。

【図9】

同実施形態におけるフロー制御動作の手順を示す図。

【図10】

同実施形態におけるフロー制御動作の第2の手順を示す図。

【図11】

同実施形態におけるエラー回復処理の手順を示す図。

【符号の説明】

2…プライマリPCIバス

4…セカンダリPCIバス

1 1…CPU

1 5…プライマリPCIシリアル転送コントローラ

3 5…セカンダリPCIシリアル転送コントローラ

1 0 0…PC本体

1 5 0, 3 5 0…コンフィグレーションレジスタ

2 0 0…ドッキングステーション

2 0 3, 3 0 3…ロックバッファ

5 0 1…ロック送信部

5 0 2…ACK待ちロック管理部

5 0 3…再送制御部

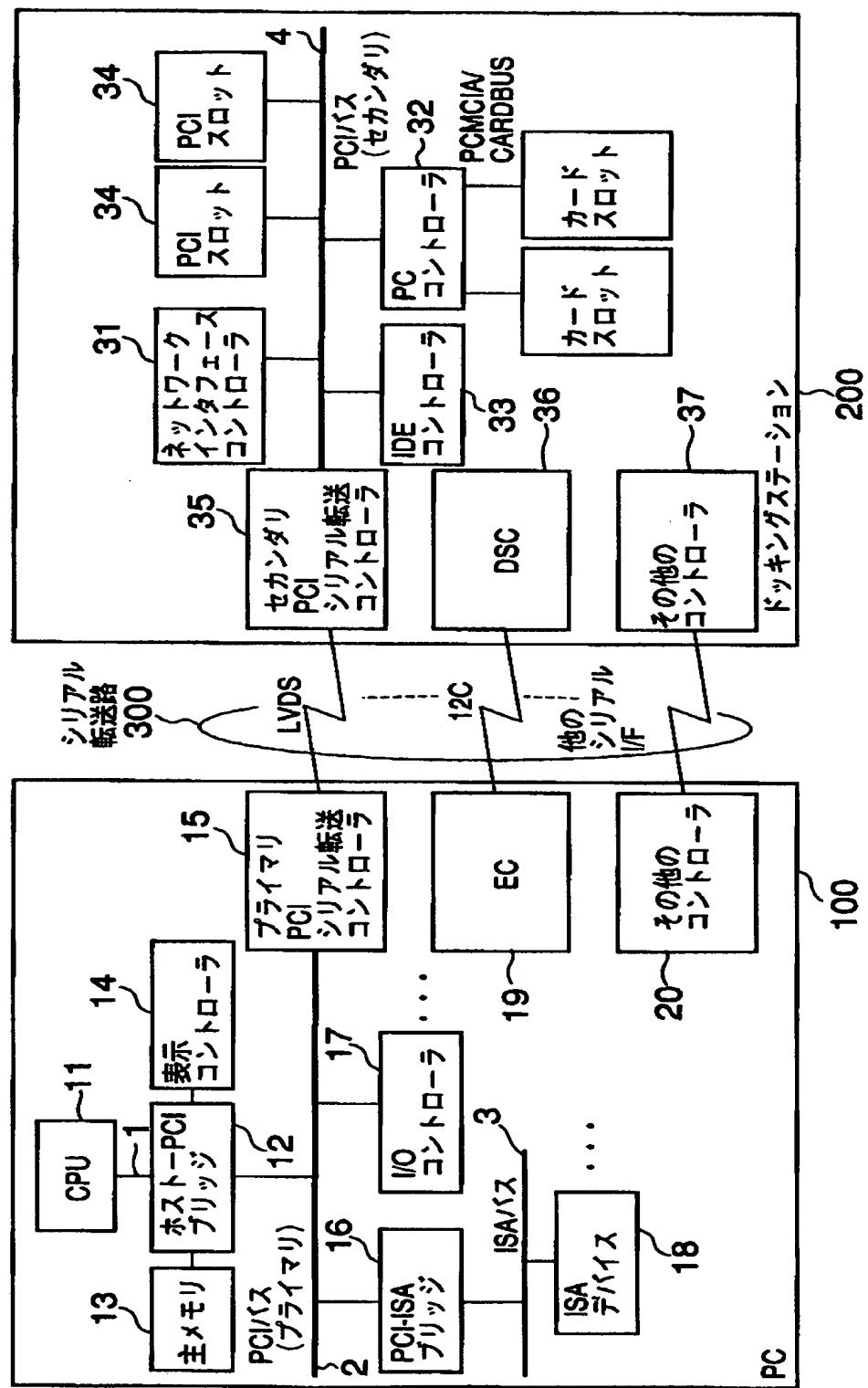
6 0 2…ACK送信部

6 0 3…NACK送信部

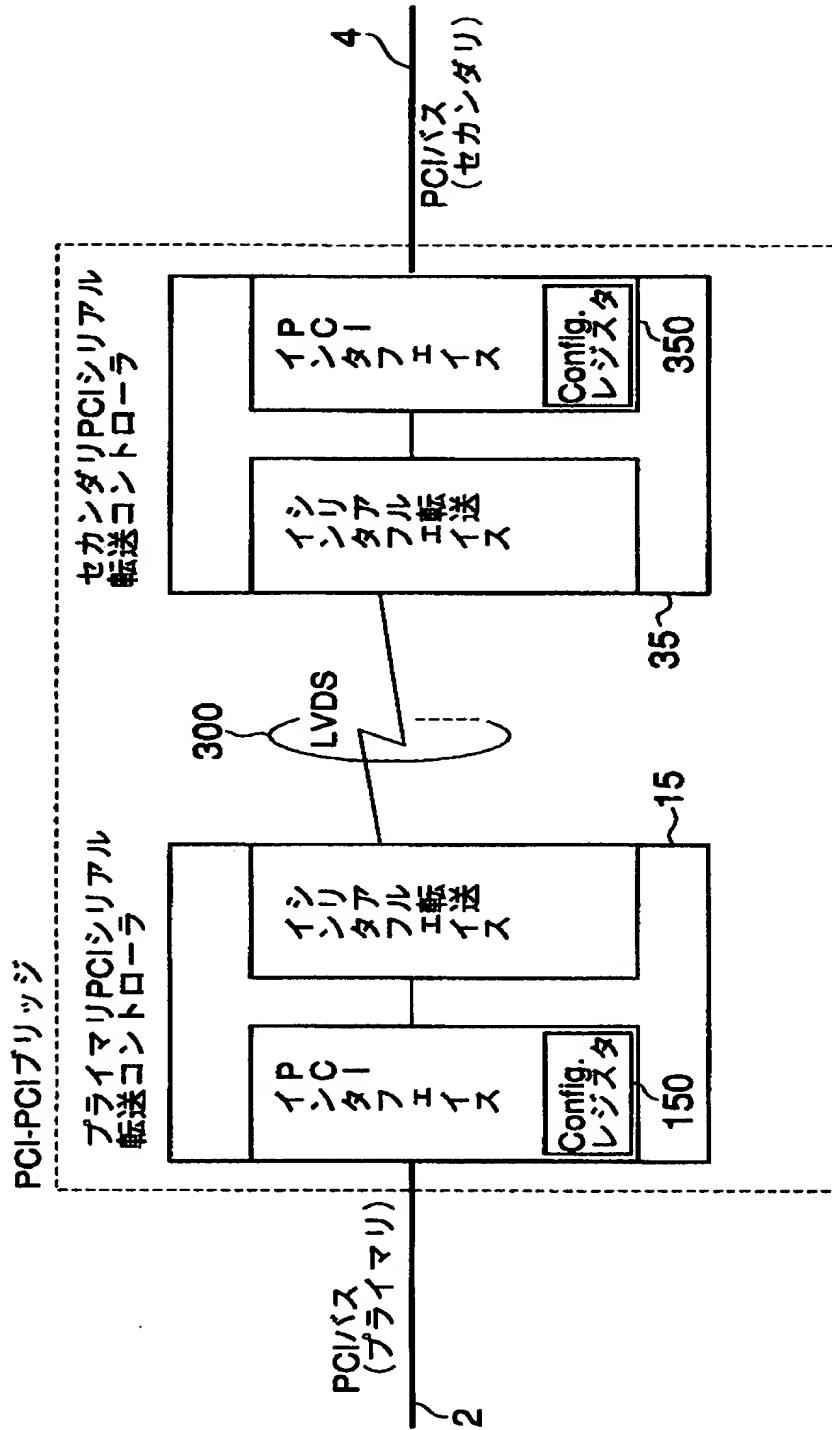
【書類名】

図面

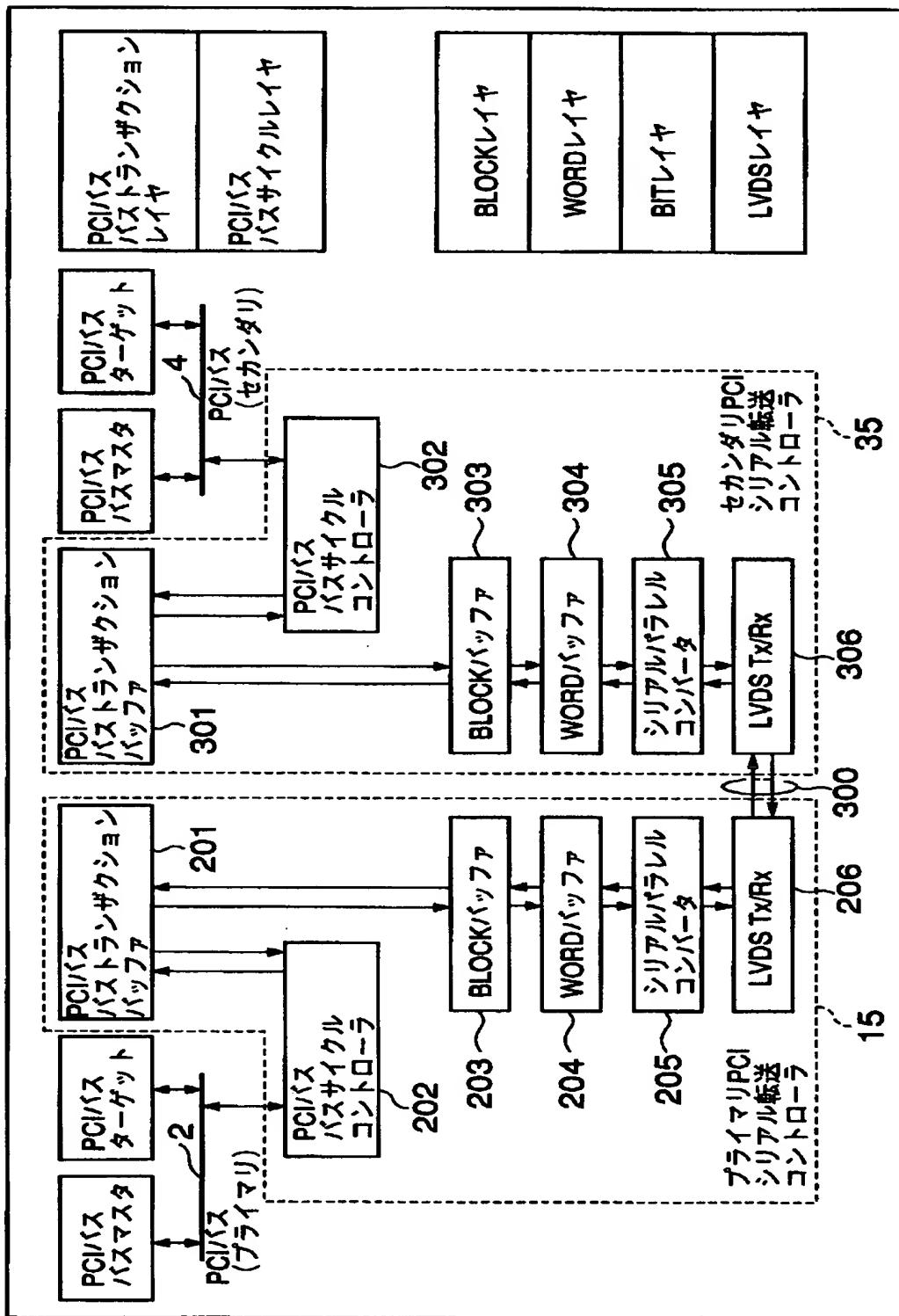
【図1】



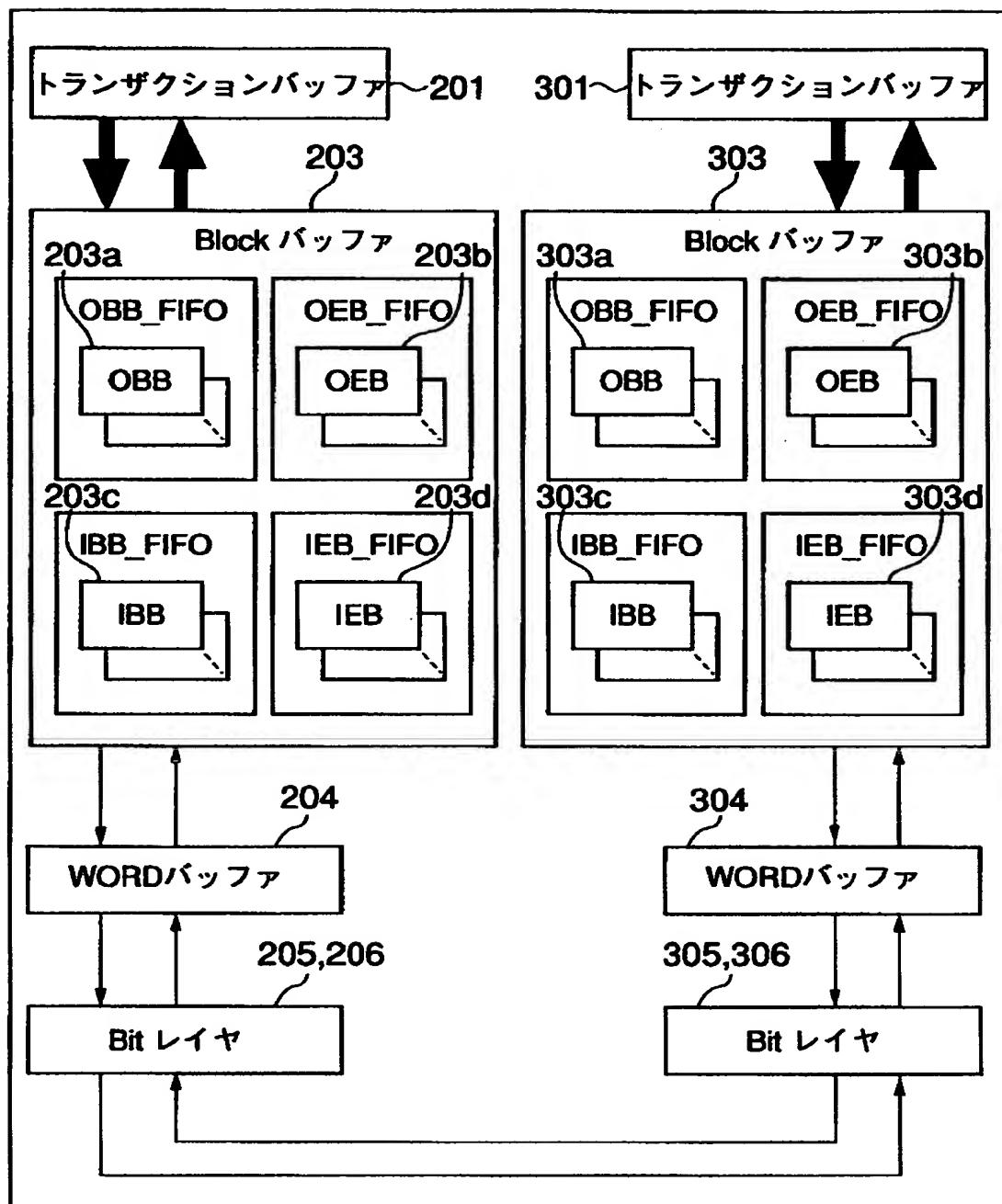
【図2】



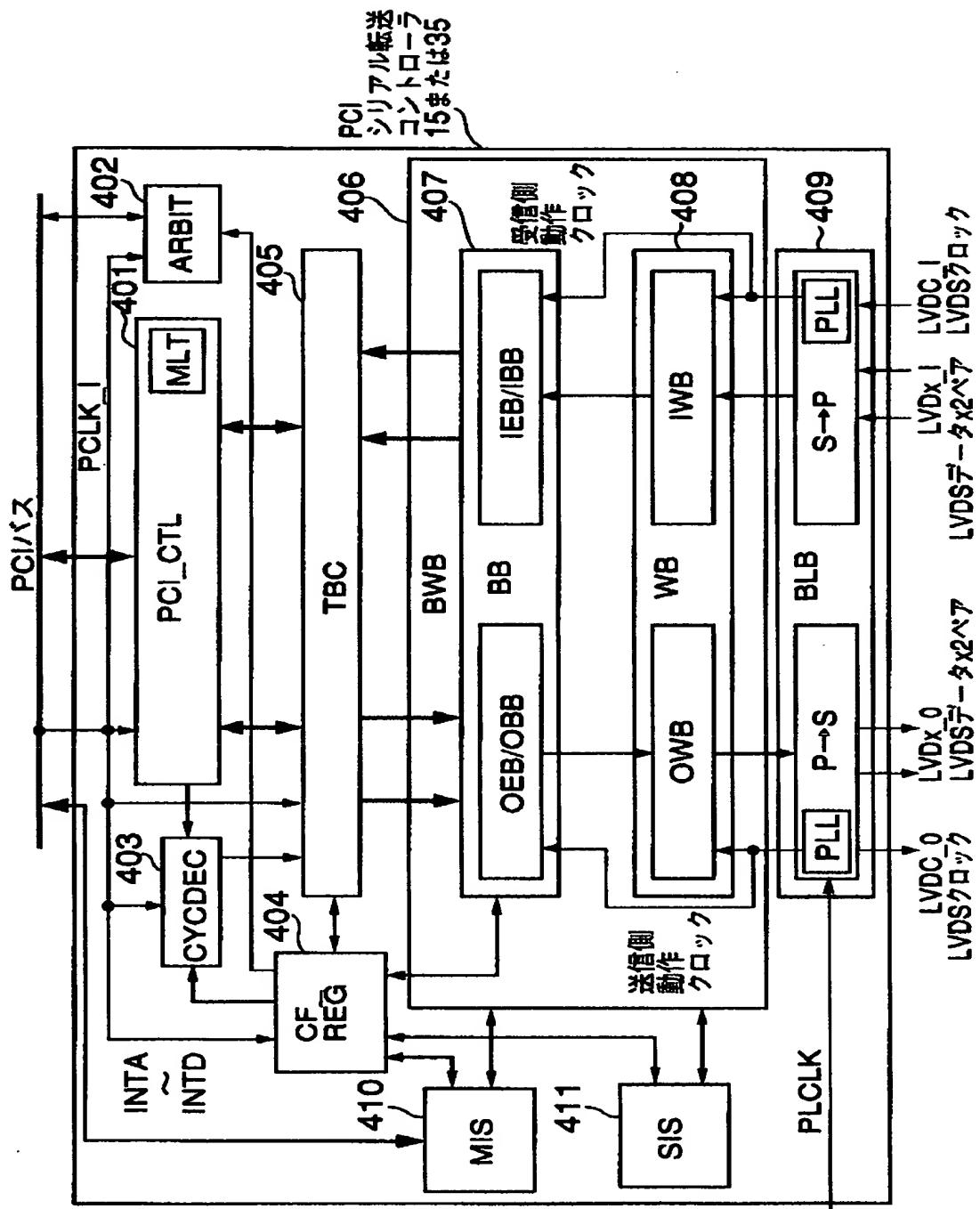
〔図3〕



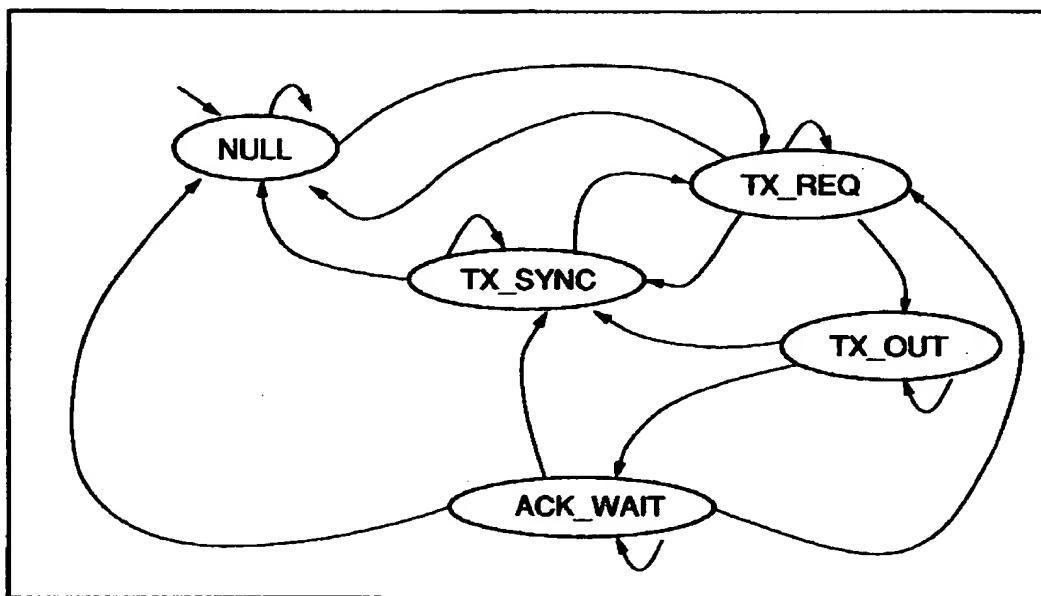
【図4】



【図5】

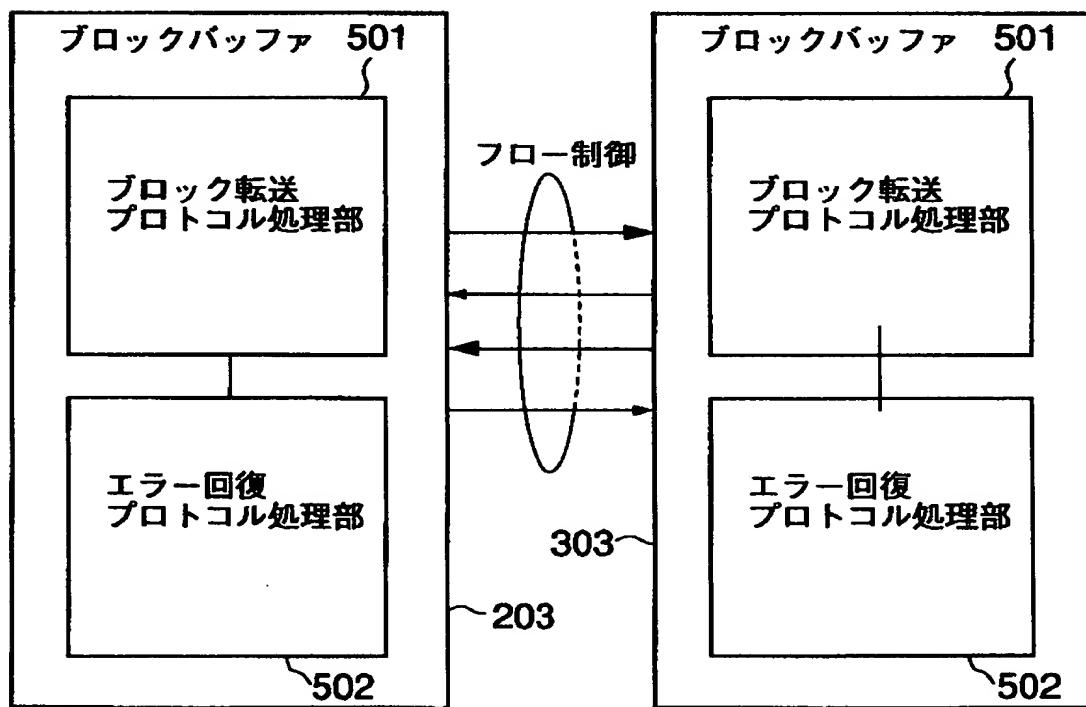


【図6】

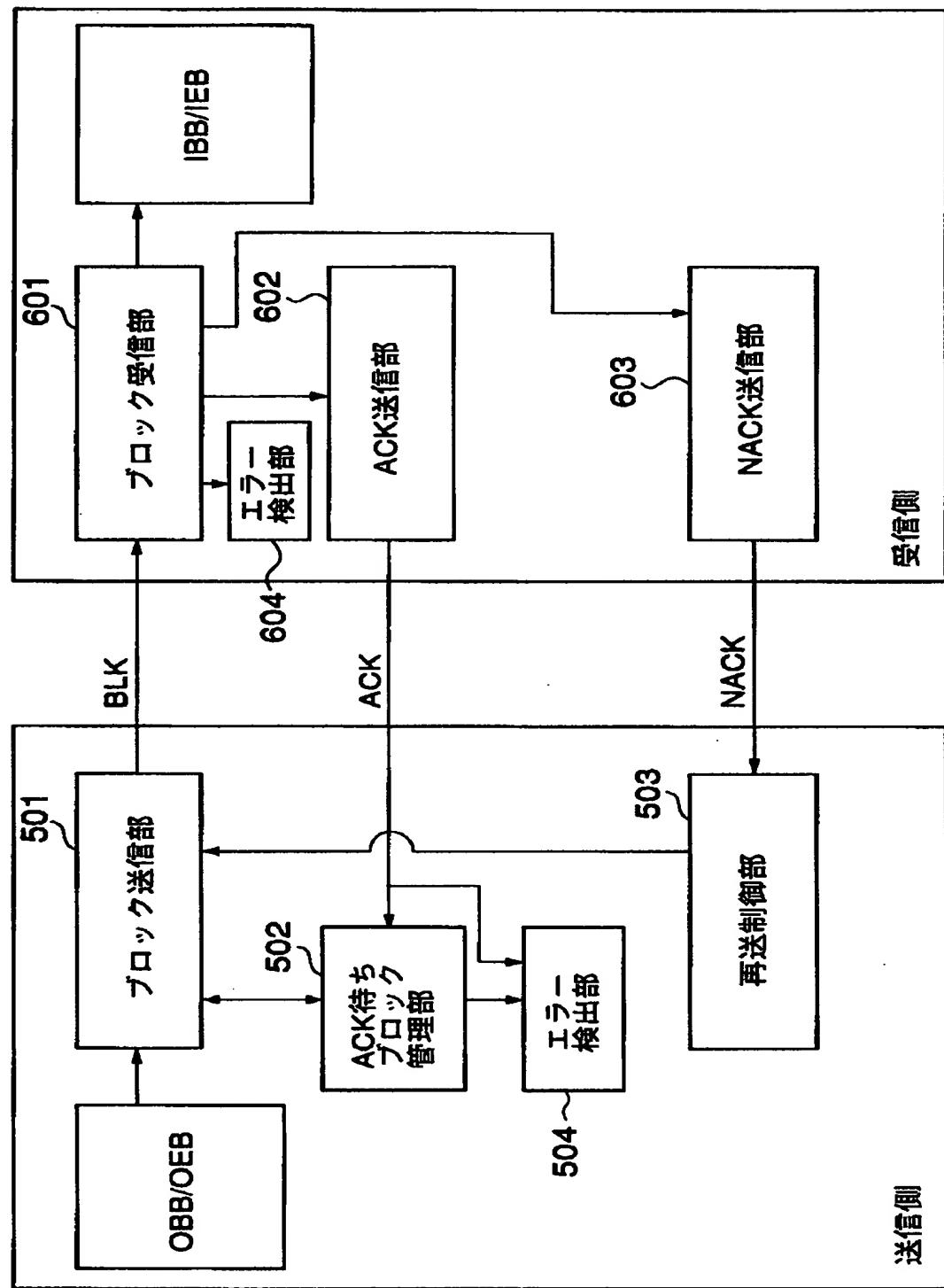


OBBの状態遷移

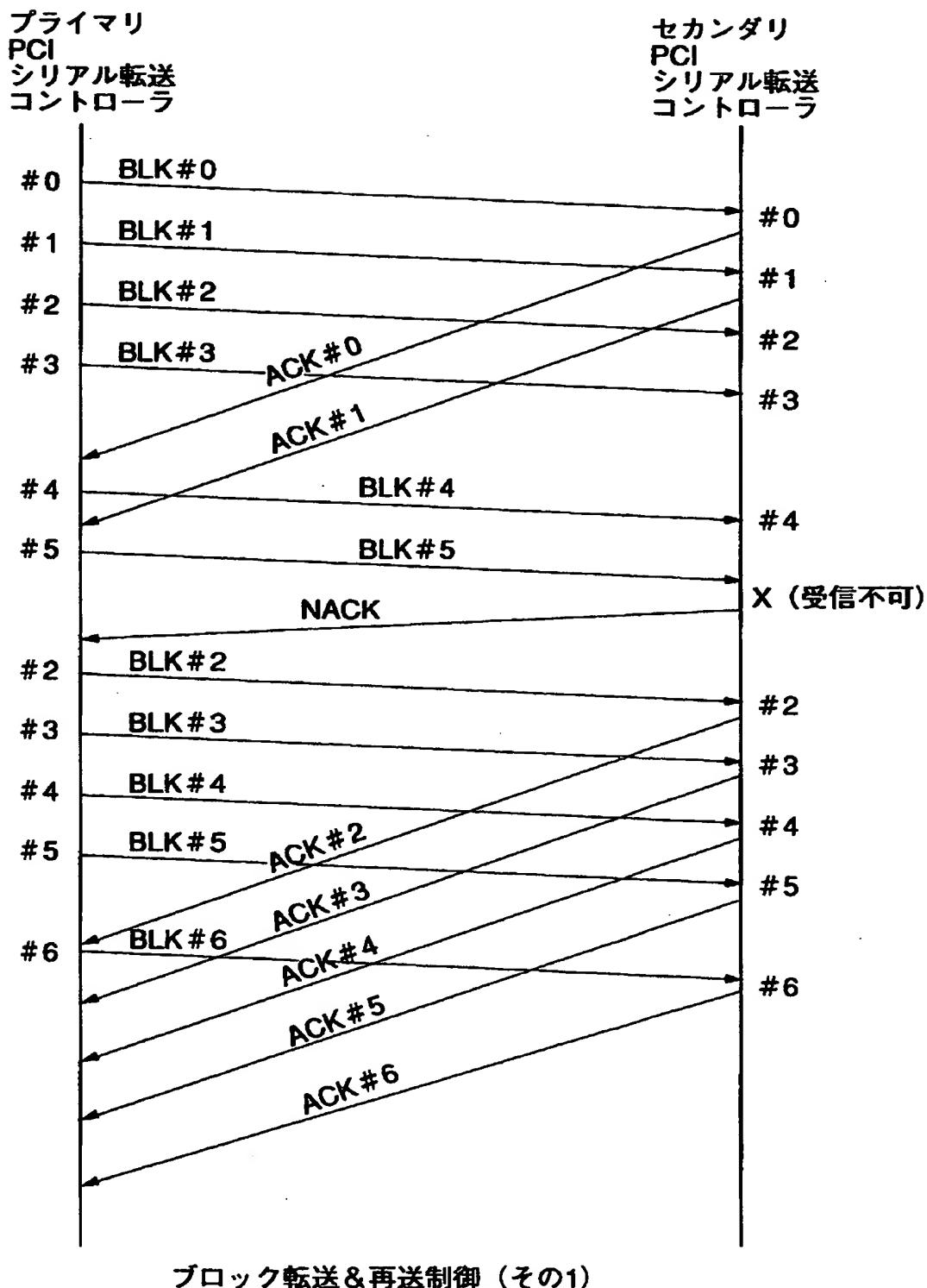
【図7】



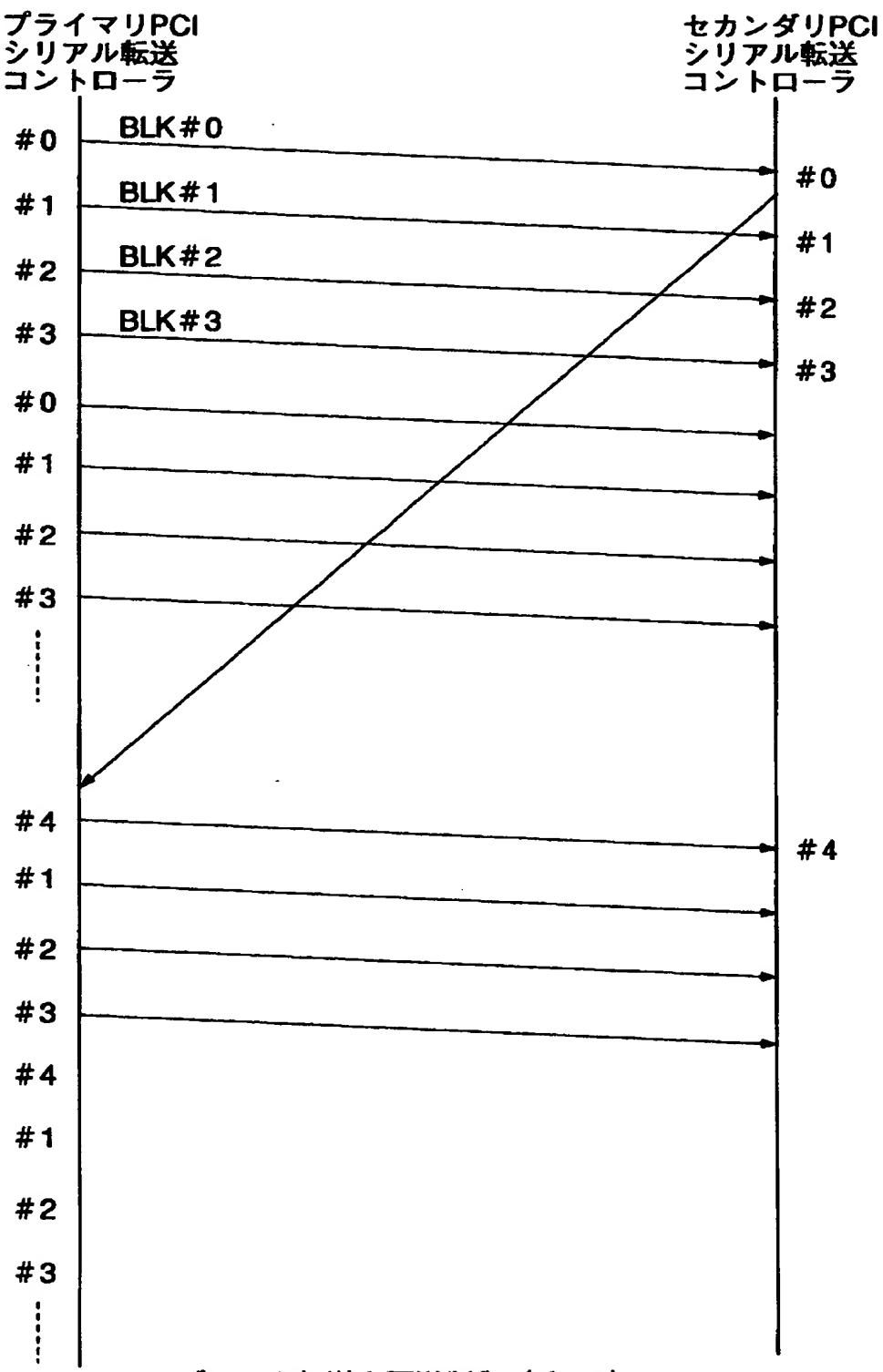
【図8】



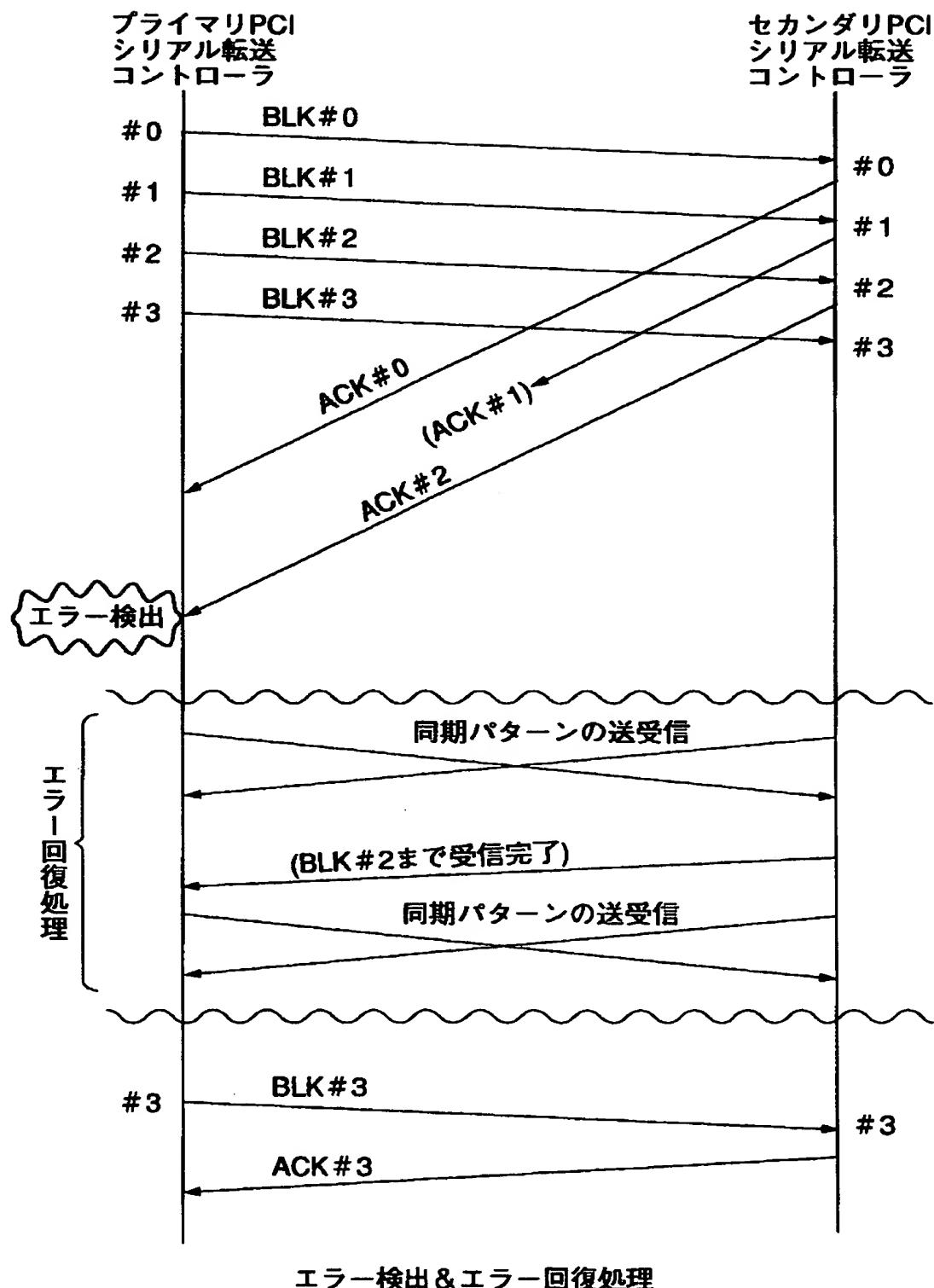
【図9】



【図10】



【図11】



【書類名】 要約書

【要約】

【課題】 バス間でトランザクションを伝達するために必要なデータを2つのコントローラ間で正確、且つ迅速に転送する。

【解決手段】 PCI-PCI ブリッジは、PC本体100側に配されたプライマリ PCIシリアル転送コントローラ15と、ドッキングステーション200側に配されたセカンダリ PCIシリアル転送コントローラ35との2つの物理的に異なるコントローラから構成される。両コントローラコントローラ15, 35間のデータ転送では、ブロックIDが付加されたブロックデータの先出し送信と、ブロックIDが付加されたACKの返信とが、非同期に行われる。送信側では、ACKに付加されたブロックIDを用いることにより、受信側がどのブロックデータの受信まで完了しているかを知ることができ、受信側からNACKを受け取ると、ACK待ちのブロックから再送を開始する。

【選択図】 図1

出願人履歴情報

識別番号 [000003078]

1. 変更年月日 1990年 8月22日

[変更理由] 新規登録

住 所 神奈川県川崎市幸区堀川町72番地
氏 名 株式会社東芝